IMPLEMENTASI APLIKASI PENGAMANAN PESAN MENGGUNAKAN ALGORITMA *DATA ENCRYPTION STANDARD (DES)* DAN *LEAST SIGNIFICANT BIT (LSB)* PADA CITRA DIGITAL

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Pada
Jurusan Teknik Informatika

Oleh :

FITRI
10651004296

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU
2011

IMPLEMENTASI APLIKASI PENGAMANAN PESAN MENGGUNAKAN ALGORITMA *DATA ENCRYPTION STANDARD* (DES) DAN *LEAST SIGNIFICANT BIT* (LSB) PADA CITRA DIGITAL

FITRI
10651004296

Tanggal Sidang : 04 Juli 2011
Periode Wisuda : November 2011

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau

ABSTRAK

Melindungi informasi yang dirahasiakan dari orang yang tidak berhak mengakses informasi tersebut diperlukan suatu cara untuk mengamankan data dan informasi. Salah satu cara pengamanan dalam pengiriman adalah merubah data menjadi yang tidak dimengerti dengan penyandian dan penyisipan menggunakan teknik kriptografi dan steganografi. Tujuan dari tugas akhir ini adalah untuk menghasilkan program aplikasi yang mampu menjaga dan memberikan keamanan yang berlapis tanpa mengurangi atau merusak pesan teks dalam citra yang disampaikan.
Aplikasi ini dibangun menggunakan algoritma *Data Encryption Standard* dan *Least Significant Bit*, algoritma ini adalah algoritma kriptografi kunci simetris berjenis *block chi*per. Algoritma penyandian citra menggunakan 64 *bit* dan 16 kali putaran berdasarkan matriks permutasi dan spesifikasi dari steganografi dilakukan dengan menggantikan bit-bit terakhir pada gambar dengan bit pesan teks.
Hasil dari penelitian ini adalah suatu aplikasi yang dapat melakukan proses enkripsi dan dekripsi pesan teks yang disisipkan dalam gambar dengan format *bitmap*.

Kata Kunci : Algoritma *Data Encryption Standard,* Citra *Bitmap,* Kriptografi, *Least Significant Bit*, Steganografi.

THE IMPLEMENTATION OF MESSAGE SECURITY USING DATA ENCRYPTION STANDARD (DES) ALGORITHM AND LEAST SIGNIFICANT BIT (LSB) ON DIGITAL IMAGE

FITRI
10651004296

Date of Final Examination : July 04th, 2011
Graduation Period : November 20011

Informatics Engineering Department
Faculty of Science and Technology
State Islamic University of Sultan Syarif Kasim Riau

ABSTRACT

To protect the confidential information from people who are not entitled to access that information requires a way to secure data and information. One way of securing in transmission is to transform the data into not understandable data by encoding and insertion using cryptography technique and steganography. The purpose of this thesis is to produce an application program that is able to maintain and provide a layered safety without reducing or damaging the text in delivered image.
This application was built using the Data Encryption Standard (DES) algorithm and Least Significant Bit, this algorithm is the symmetric key cryptographic algorithm with block chiper type. The image encryption algorithm using 64 bit and 16 times iteration based on the permutation matrix and specification of steganography is done with replase the last bits in the images with bits in texts.
The outcome of this research is an application which can perform encryption process and decryption the inserted text in the image with bitmap format.

Keywords *: Algorithm Data Encryption Standard, Bitmap Image, Cryptography, Least Significant Bit, Steganography.*

DAFTAR ISI

Halaman

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat ini berbagai macam teknik digunakan untuk melindungi informasi yang dirahasiakan dari orang yang tidak berhak terhadap hak akses informasi tersebut, maka diperlukan suatu cara untuk mengamankan data dan informasi. Salah satunya adalah dengan cara merubah data tersebut ke dalam bentuk data yang lain yang tidak dapat dimengerti dalam bentuk penyandian dan penyisipan kerahasiaan data dengan teknik kriptografi dan steganografi.

Dalam kriptografi, pesan yang tersimpan disandikan dalam bentuk yang tidak dapat dipahami agar makna pesan tidak dimengerti oleh pihak lain dengan aspek keamanan seperti kerahasiaan, integritas data, serta autentikasi. Algoritma kriptografi yang paling terkenal dan populer adalah algoritma *Data Encryption Standard* (DES). DES merupakan algoritma chiper blok yang ditetapkan sebagai standar nasinonal yang paling banyak dipakai di dunia, yang di adopsi oleh NIST (*National Institute of Standards and Technology*). Algoritma DES telah mendapatkan persetujuan dari *National Bureau of Standard* (NBS) setelah penilaian kekuatannya oleh *National Security Agency* (NSA) Amerika Serikat. Teknik pengamanan data yang lain adalah dengan menggunakan metode steganografi. Spesifikasi dari steganografi adalah berfungsi untuk menyamarkan eksistensi data rahasia sehingga sulit dideteksi. Metode steganografi yang digunakan adalah metode *Least Significant Bit* (LSB). Metode ini merupakan penyembunyian pesan yang dilakukan mengganti bit-bit data yang kurang berarti dalam segmen citra dengan bit-bit rahasia pada bit terakhir sehingga tidak berpengaruh terhadap persepsi *visual*/auditori.

Pada penelitian sebelumnya, memaparkan teknik LSB yang merupakan teknik pelengkap dari kriptografi yang dapat memberikan ketahanan keamanan data (Setiawan, 2009). Yang memiliki suatu objek penyisipan data berupa citra

digital, *audio*, dan *video.* Pada citra, format yang paling banyak digunakan adalah *bitmap* 24 bit yang dikarenakan merupakan suatu representasi dari citra grafis yang terdiri dari susunan titik yang tersimpan di memori komputer dan file gambar yang dapat dibuka di semua program pengolah gambar, disamping itu ukuran *byte*-nya dapat dihitung tinggi dan lebarnya dalam pixel sehingga sangat mudah untuk disandikan (Nugroho dkk, 2007). Kesimpulan ini didukung dengan penelitian Stefanus Astrianto (2007).

Berdasarkan defisiensi studi penelitian sebelumnya diperoleh penjelasan bahwa pesan yang disampaikan melalui proses sisipan citra bitmap tidak begitu aman, sehingga untuk meningkatkan keamanan pesan tersebut dilakukan proses teknik kriptografi. Pesan mengalami perubahan menjadi yang tidak dimengerti (Rehazain, 2007). Sehingga hal tersebut yang menjadi dasar penelitian pada tugas akhir ini.

Keamanan data dikombinasikan dengan teknik pengamanan data kriptografi dan teknik steganografi yang menjadikan data bersifat rahasia. Penggabungan ini akan memberikan keamanan berlapis sebagai tanda kepemilikan tanpa mengurangi atau merusak informasi pesan tersebut. Pesan yang sudah dienkripsi disisipkan, sehingga memberikan *invisibility* yang cukup tinggi pada citra *bitmap* oleh kasat mata manusia.

Pada penelitian ini penulis akan membahas lebih lanjut tentang implementasi aplikasi pengamanan pesan menggunakan algoritma *Data Encryption Standard* (DES) dan *Least Significant Bit* (LSB) pada citra bitmap.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, permasalahan yang dapat dirumuskan pada tugas akhir ini, adalah bagaimana mengimplementasikan suatu aplikasi pengamanan pesan menggunakan algoritma *Data Encryption Standart* (DES) dan *Least Significant Bit* (LSB) pada citra digital.

1.3 Batasan Masalah

Adapun batasan masalah dalam tugas akhir ini adalah sebagai berikut :

1. Media objek pengamanan pesan adalah citra berformat *bitmap* 24 bit.
2. Pesan yang dienkripsi lalu disisipkan adalah berupa teks bukan file.
3. Penelitian tidak secara mendalam membahas masalah kualitas dan ketahanan citra.
4. Aplikasi ini tidak membahas proses pengiriman dan penerimaan data, hanya mencakup implemetasi enkripsi dan dekripsi serta penyisipan.

1.4 Tujuan

Tujuan yang ingin dicapai dalam penyusunan tugas akhir ini adalah dapat membangun aplikasi pengamanan pesan pada citra digital dengan menggunakan algoritma *Data Encryption Standard* (*DES)* dan dengan metode *Least Significant Bit (LSB).*

1.5 Sistematika Penulisan

Sistematika penulisan dalam penyusunan laporan tugas akhir ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan dasar-dasar dari penulisan laporan tugas akhir, yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan, serta sistematika penulisan laporan tugas akhir.

BAB II LANDASAN TEORI

Bab ini membahas teori-teori yang berhubungan dengan topik penelitian yang meliputi keamanan data, kriptografi, algoritma *Data Encryption Standard* (*DES)*, steganografi, *Least Significant Bit Insertion (LSB),* dan citra digital *bitmap*.

BAB III METODOLOGI PENELITIAN

Bab ini membahas tentang metodologi yang digunakan dalam penelitian dan pengembangan perangkat lunak.

BAB IV ANALISA DAN PERANCANGAN

Pada bab ini merupakan pembahasan tentang analisis perangkat lunak, meliputi analisa, analisa masalah, analisa metode, analisa kebutuhan sistem, serta perancangan. Perancangan sistem memiliki rancangan prosedural yang terdiri dari perancangan diagram alir (*flowchart*), *context diagram*, dan *data flow diagram* (DFD) .

BAB V IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas implementasi dan pengujian yang dilakukan terhadap aplikasi pengamanan pesan dengan algoritma *Data Encryption Standard* (*DES)* dan dengan metode *Least Significant Bit* (LSB) pada citra digital.

BAB VI PENUTUP

Bab ini berisi kesimpulan yang dihasikan dari pembahasan tentang implementasi pengamanan pesan menggunakan algoritma *Data Encryption Standard* (*DES)* dan *Least Significant Bit* (LSB) pada citra digital dan saran sebagai hasil akhir dari penelitian yang telah dilakukan.

BAB II

LANDASAN TEORI

Landasan teori disusun berdasarkan teori-teori mengenai metode yang digunakan dalam keamanan data, kriptografi, algoritma *Data Encryption Standard (DES),* steganografi, *Least Significant Bit (LSB),* dan citra digital.

2.1 Keamanan Data

Keamanan dan kerahasiaan data merupakan hal yang sangat penting yang terus berkembang, karena memiliki peranan yang sangat besar dalam sistem komputer, jaringan komputer, dan penggunaan teknologi komputer. Beberapa kasus menyangkut keamanan data saat ini menjadi sesuatu yang membutuhkan biaya penanganan dan pengamanan yang sedemikian besar.

Dalam mengatasi masalah keamanan data tersebut, menurut *International Telecomunication Union-Telecomunication Standarization Sector (ITU-T)* yang merekomendasikan X.8000 ada beberapa teknik dalam menjaga keamanan dan kerahasiaan dalam sistem komputer, jaringan komputer, dan *internet*, yaitu: (Budi Raharjo,1998).

1. *Privacy*
2. *Integrity*
3. *Authencity*
4. *Non-Repudiation*

2.2 Kriptografi

Kriptografi berasal dari dua kata bahasa Yunani, *Cryptos* dan *Graphein*. *Cryptos* berarti *secret* (rahasia) dan *Graphia* berarti *writing* (tulisan). Jadi, kriptografi berarti ”*secret writing*” (tulisan rahasia). Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lain. (Ariyus,2008).

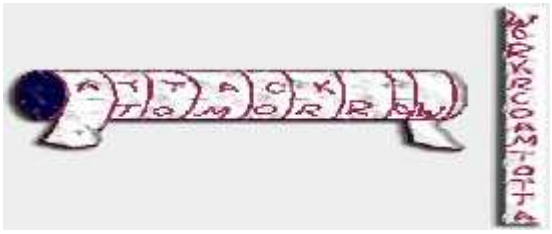
Kriptografi adalah ilmu dan seni dalam mengamankan pesan dengan cara mengubah pesan menjadi sesuatu yang tidak dapat dimerngerti oleh orang lain dengan teknik-teknik dan metode-metode tertentu. Kriptografi tidak berarti hanya memberikan keamanan informasi saja, namun lebih ke arah teknik-tekniknya. Algoritma kriptografi yang baik tidak ditentukan oleh kerumitan dalam mengolah data atau pesan yang akan disampaikan. Ada empat syarat yang perlu dipenuhi dalam ilmu kriptografi (sebagai aspek-aspek keamanan), yaitu (Munir,2006) :

1. Kerahasiaan (*confidentiality*), adalah menyediakan privasi untuk pesan dan menyimpan data dengan menyembunyikannya. Pesan *(plaintext)* hanya dapat dibaca oleh pihak yang memiliki kewenangan, yang direalisasikan dengan menyandikan pesan menjadi chiperteks.
2. Integritas data (*data integrity*) , berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain menyangkut penyisipan, penghapusan, dan pensubtitusian data lain ke dalam data yang sebenarnya, yang tidak dimodifikasi ketika sedang dalam proses transmisi data.
3. Autentikasi (*authentication*), beruhubunagn dengan identifikasi, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi (*user/entity authentification*)maupun mengidentifikasi kebenaran sumber pesan (*data origin authentification*),agar penerima informasi dapat memastikan keaslian pesan, bahwa pesan itu datang dari orang yang dimintai informasi. dengan kata lain, informasi itu benar-benar datang dari orang yang dikehendaki. Pengirim pesan harus dapat diindentifikasi dengan pasti.
4. Non-repudiasi (*non-repudiation*), yang berarti dapat membuktikan bahwa dokumen memang benar datang dari orang yang dimintai informasi. Misalnya X dan X tidak menyangkalnya.

2.2.1. Sejarah Kriptografi

Kriptografi mempunyai sejarah yang sangat menarik dan panjang. kriptografi sudah digunakan 4000 tahun yang lalu, diperkenalkan oleh orang-orang Mesir lewat *hieroglyph.* Jenis tulisan ini bukanlah bentuk standar untuk menulis pesan. (Ariyus,2008)

Sekitar tahun 400 SM *chiper* transposisi digunakan oleh bangsa spartan di Yunani. Mereka menggunkan alat yang diberi nama *scytel* yang terdiri dari sebuah kertas panjang dari daun *papyrus* yang dililitkan pada sebuah silinder dengan diameter tertentu yang merupakan kunci penyandian. Pesan ditulis secara horizontal, baris perbaris. Bila pita dilepaskan, maka huruf-huruf di dalamnya telah tersusun secara acak membentuk pesan rahasia. Untuk membaca pesan, penerima pesan harus melilitkan kembali kertas tersebut ke silinder yang memiliki diameter sama dengan kunci penyandian



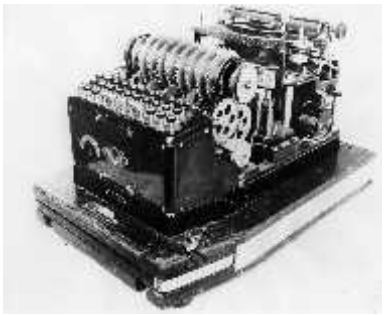
Gambar 2.1 *Scytel*

Sedangkan *chiper* subtitusi paling awal dan paling sederhana pernah dipergunakan pada zaman Romawi kuno oleh raja Yunani kuno Julius Caesar. Enkripsi dilakukan dengan menggeser karakter-karakter dalam abjad. Jika enkripsi dilakukan dengan menggeser 13 huruf ke kanan, maka huruf A disandikan dengan N, huruf B dengan O dan seterusnya.

Gambar 2.2 Ilustrasi Caesar *Chiper*

Sejarah juga mencatat bahwa kriptografi digunakan untuk tujuan keagamaan. Kalangan gereja pada masa awal agama Kristen menggunakan kriptografi untuk menjaga tulisan relijius dari gangguan otoritas politik atau budaya yang dominan pada saat itu. Mungkin yang sangat terkenal adalah *"number of the Beast"* di dalam Kitab Perjanjian Baru. Angka "666" menyatakan cara kriptografik (yaitu dienkripsi) untuk menyembunyikan pesan berbahaya (Munir, 2006).

Pada masa perang dunia ke II, tentara Nazi Jerman menggunakan mesin *Enigma* atau juga disebut dengan mesin *rotor* yang digunakan Hitler untuk mengirimkan pesan kepada tentaranya di medan perang. Masin yang menggunakan beberapa buah *rotor* (roda berputar) ini melakukan enkripsi dengan cara yang cukup rumit. Namun, *Enigma chiper* berhasil dipecahkan oleh tentara sekutu yang merupakan faktor sehingga memperpendek perang dunia ke II. Setelah Jerman mengetahui bahwa *enigma* dapat dipecahkan, maka enigma mengalami beberapa kali perubahan. *Enigma* yang digunakan Jerman dapat mengenkripsi suatu pesan sehingga mempunyai 15×1018 kemungkinan untuk dapat mendekripsi pesan.(Munir, 2006)



Gambar 2.3. Mesin Enigma

Perkembangan komputer dan sistem komunikasi pada tahun 1960-an mengakibatkan munculnya kebutuhan pihak swasta akan alat untuk melindungi informasi dalam bentuk digital dan untuk menyediakan layanan keamanan informasi. Kriptografi digital dimulai pada tahun 1970 atas usaha *Feistel* dari IBM dan memuncak pada tahun 1977 dengan diadopsinya sistem kriptografi DES (*Data Encryption Standard*) oleh *U.S. Federal Information Processing Standard*

untuk mengenkripsi informasi rahasia. DES merupakan mekanisme kriptografi yang paling terkenal dalam sejarah dan tetap menjadi standar pengamanan data elektronik komersial pada kebanyakan institusi keuangan di seluruh dunia.

Sampai pada akhir Perang Dunia I, kriptografi merupakan disiplin ilmu matematika yang spesial. Penelitian dalam bidang ini tidak pernah sampai kepada umum sehingga tidaklah mengherankan kalau banyak orang tidak mengetahui keberadaan ataupun manfaat darinya.

Pengembangan paling mengejutkan dalam sejarah kriptografi terjadi pada 1976 saat Diffie dan Hellman mempublikasikan "*New Directions in Cryptography*". Tulisan ini memperkenalkan konsep revolusioner kriptografi kunci publik dan juga memberikan metode baru untuk pertukaran kunci, keamanan yang berdasar pada kekuatan masalah algoritma diskrit. Meskipun Diffie dan Hellman tidak memiliki realisasi praktis pada ide enkripsi kunci publik saat itu, idenya sangat jelas dan menumbuhkan ketertarikan yang luas pada komunitas kriptografi.

Algoritma kriptografi modern tidak lagi mengandalkan keamanannya pada kerahasiaan algoritma tetapi kerahasiaan kunci. *Plaintext* yang sama bila disandikan dengan kunci yang berbeda akan menghasilkan *chipertext* yang berbeda pula. Dengan demikian algoritma kriptografi dapat bersifat umum dan boleh diketahui oleh siapa saja, akan tetapi tanpa pengetahuan tentang kunci, data tersandi tetap saja tidak dapat terpecahkan.

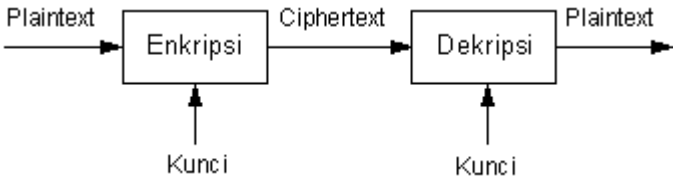
2.2.2 Algoritma Kriptografi

Algoritma kriptografi terdiri dari tiga aspek dasar, yaitu:

1. Enkripsi merupakan hal yang sangat penting dalam kriptografi, merupakan pengamanan data yang dikirimkan agar terjaga kerahasiaannya. Pesan asli disebut *plaintext*, yang di ubah menjadi kode-kode yang tidak dimengerti atau disebut *chipertext*.

2. Dekripsi merupakan kebalikan dari proses enkripsi. Pesan yang enkripsi dikembalikan ke bentuk asalnya atau dengan kata lain proses membalikkan *chipertext* menjadi *plaintext*.
3. Kunci yang dimaksud disini adalah kunci yang dipakai untuk melakukan enkripsi dan dekripsi. Kunci terbagi menjadi dua bagian, kunci rahasia (*private key)* dan kunci umum *(Public key).*

Ilustrasi untuk menggambarkan kondisi kriptografi dapat dilihat pada gambar di bawah:



Gambar 2.4 Proses Enkripsi dan Dekripsi (*Munir,2004*)

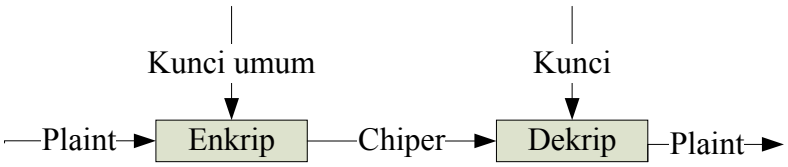
2.2.3 Jenis Algoritma Kriptografi

Berdasarkan kunci yang dipakai, algoritma kriptografi dapat dibedakan atas dua golongan, yaitu :

1. *Asymmetric Algorithms*
2. *Symmetric Algorithms*

2.2.3.1 *Asymmetric Algorithms*

Asymmetric Algorithms adalah algoritma yang menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsinya. Algoritma ini disebut juga algoritma kunci umum (*public key algorithm*) karena kunci untuk enkripsi dibuat umum (*public key*) atau dapat diketahui oleh setiap orang, tapi kunci untuk dekripsi hanya diketahui oleh orang yang berwenang mengetahui data yang disandikan atau sering disebut kunci pribadi (*private key*). Proses enkripsi-dekripsi algoritma asimetris dapat dilihat pada gambar dibawah ini :



Gambar 2.5 Proses Enkripsi dan Dekripsi Algoritma Asimetris (*Munir,2004*)

Pada algoritma *public key* ini, semua orang dapat mengenkripsi data dengan memakai *public key* penerima yang telah diketahui secara umum. Akan tetapi data yang telah terenkripsi tersebut hanya dapat didekripsi dengan menggunakan *private key* yang hanya diketahui oleh penerima. Beberapa contoh algoritma kunci publik antara lain :

1. DSA (*Digital Signature Algorithm*).
2. RSA
3. LUC

1. DSA

Pada bulan Agustus 1991, NIST (*The National Institute of Standard and Technology*) mengumumkan algoritma sidik dijital yang disebut *Digital Signature Algorithm* (DSA). DSA dijadikan sebagai standar dari *Digital Signature Standard* (DSS). DSA menggunakan fungsi *hash* SHA (*Secure Hash Algorithm*) untuk mengubah pesan menjadi *message digest* yang berukuran 160 *bit*.

DSA tidak dapat digunakan untuk enkripsi. DSA mempunyai dua fungsi utama:

1. Pembentukan sidik digital (*signature generation*)
2. Pemeriksaan keabsahan sidik digital (*signature verification*).

2. RSA

Algoritma RSA dibuat oleh 3 orang peneliti dari MIT (*Massachussets Institute of Technology*) pada tahun 1976, yaitu: Ron (R)ivest, Adi (S)hamir, dan Leonard (A)dleman.

Dalam pemakaian sistem penyandian RSA dalam kehidupan sehari-hari adalah *signature* atau tanda tangan *digital* dalam surat elektronik dan untuk autentikasi sebuah data. Untuk meyakinkan penerima surat elektronik yang ditandatangani, diperlukan pembuktian bahwa surat elektronik tersebut memang berasal dari sipengirim.

3. LUC

Algoritma LUC merupakan algoritma kriptografi kunci public yang dikembangkan oleh Peter J. Smith dari New Zealand pada tahun 1993. Metode LUC ini dirancang oleh Peter J. Smith setelah ia berhasil meneliti dan melihat kelemahan dari metode RSA. Metode LUC ini menggunakan fungsi Lucas yang dapat menutupi kelemahan metode RSA yang menggunakan fungsi pangkat. Kemungkinan untuk menjebol RSA menjadi ada karena masalah pangkat tersebut. Fungsi Lucas ini dapat mencegah kemungkinan tersebut.

2.2.3.2 *Symmetric Algorithms*

Algoritma kriptografi simetris atau disebut juga algoritma kriptografi konvensioanl adalah algoritma yang menggunakan kunci untuk proses enkripsi sama dengan kunci untuk proses dekripsi.

Algoritma kriptografi simetri dapat dikelompokkan menjadi dua kategori, yaitu:

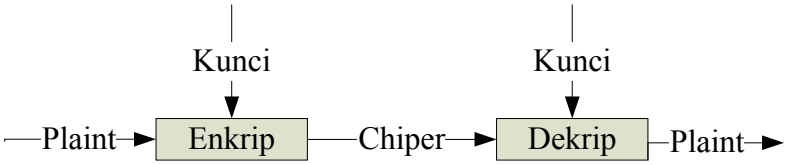
a. *Chiper* aliran (s*tream chiper*)

Algoritma kriptografi beroperasi pada plainteks/chiperteks dalam bentuk *bit* tunggal, pada rangkaian *bit* ini dienkripsikan dan didekripsikan *bit* per *bit*.

b. *Chiper* blok (*block chiper*)

Algoritma kriptografi beroperasi pada plainteks/chiperteks dalam bentuk blok *bit*, yang dalam hal ini rangkaian *bit* dibagi menjadi blok-blok *bit* yang panjangnya sudah ditentukan sebelumnya.

Proses enkripsi dan dekripsi algoritma kriptografi simetris dapat dilihat pada gambar dibawah ini :



Gambar 2.6 Proses Enkripsi dan Dekripsi Algoritma Simetris (*Munir,2004*)

Algoritma – algoritma yang termasuk dalam *chiper block*, yaitu :

1. Algoritma DES *(Data Encryption Standard)*
2. Algoritma *Triple DES (Data Encryption Standard)*
3. Algoritma IDEA
4. Algoritma *Blowfish*
5. Algoritma *Rinjdael*
6. Algoritma *Twofish*

1. Algoritma DES *(Data Encryption Standard)*

Algoritma DES merupakan kunci simetris, dikembangkan di IBM dibawah kepemimpinan W.L. Tuchman pada tahun 1972. Algoritma ini didasarkan pada algoritma LUCIFER yang dibuat oleh Horst Feistel. Algoritma DES telah mendapat persetujuan dari *National Bureau of Standard* (NBS) setelah penilaian kekuatannya oleh *National Security Agency* (NSA) Amerika Serikat.

2. Algoritma *Triple DES (Data Encryption Standard)*

Algoritma *Triple DES* dikenal juga dengan TDEA *(Triple Data Encryption Algorithm)*, yang merupakan algoritma kriptografi simetri yang beroperasi dalam bentuk blok. *Triple DES* mengaplikasikan *DES* sebanyak 3 kali, yang dikarenakan *DES* dianggap sudah tidak aman lagi, yang perangkat keras khususnya kuncinya dapat ditemukan dalam waktu beberapa hari. Kemudian IBM yang membuat algoritma *DES* mengembangkannya menjadi *triple DES*. *Triple DES* juga banyak digunakan dan penggunaanya lebih aman dibandingkan *DES*. Namun pengenkripsinnya tidak mengubah algoritma dari *DES.*

3. Algoritma IDEA

Algoritma IDEA (*International Data Encryption Algorithm*) merupakan sebuah algoritma kunci simetrik (*Symmetric Algorithms*). IDEA muncul pertama kali pada tahun 1990 yang dikembangkan oleh ilmuwan Xueijia Lai dan James L Massey.

IDEA mengenkripsi *plaintext* menjadi chiperteks yang lemah hanya mengalami 8 putaran. Algoritma ini membagi *plaintext* yang akan dienkripsi menjadi 4 blok, masing-masing terdiri dari 16 *bit*, tetapi IDEA lebih meningkatkan proses keamanan kunci, dimana 52 upa kunci (sub-keys) yang terdiri dari 16 *bit* dibangkitkan dari kunci utama (master key) yang terdiri 128 *bit*. Lalu pada setiap putarannya digunakan 6 kunci. Setelah itu dilakukan transformasi final dengan 4 kunci untuk membalikkan posisi ke operasi dasar.

4. Algoritma *Blowfish*

Blowfish merupakan sebuah algoritma kunci simetrik (*symmetric Algorithms) chiper* blok yang dirancang pada tahun 1993 oleh Bruce Schneier. Pada saat itu banyak sekali rancangan algoritma yang ditawarkan, namun hampir semua terhalang oleh paten atau kerahasiaan pemerintah Amerika. Schneier menyatakan bahwa *blowfish* bebas paten dan akan berada pada domain publik.

Keberhasilan *blowfish* dalam menembus pasar telah terbukti dengan diadopsinya *blowfish* sebagai *Open Cryptography Interface* (OCI) pada *kernel Linux versi* 2.5 keatas. Dengan diadopsinya *blowfish*, maka telah menyatakan bahwa dunia *open source* menganggap *blowfish* adalah salah satu algoritma yang terbaik.

5. Algoritma *Rinjdael*

Seperti pada *DES*, *Rijndael* menggunakan substitusi dan permutasi, dan sejumlah putaran (*cipher* berulang) – setiap putaran mengunakan kunci internal yang berbeda (kunci setiap putaran disebut *round key*). Tetapi tidak seperti *DES* yang berorientasi bit, *Rijndael* beroperasi dalam orientasi *byte* (untuk memangkuskan implementasi algoritma ke dalam *software* dan *hardware*). Garis

besar Algoritma *Rijndael* yang beroperasi pada blok 128-bit dengan kunci 128-bit adalah sebagai berikut (di luar proses pembangkitan *round key*):

1. *AddRoundKey*: melakukan *XOR* antara *state* awal (plainteks) dengan *cipher key*. Tahap ini disebut juga *initial round*.
2. Putaran sebanyak *Nr* – 1 kali. Proses yang dilakukan pada setiap putaran adalah:
   a. *SubBytes*: substitusi *byte* dengan menggunakan table substitusi (*S-box*).
   b. *ShiftRows*: pergeseran baris-baris *array state* secara *wrapping*.
   c. *MixColumns*: mengacak data di masing-masing kolom *array state.*
   d. *AddRoundKey*: melakukan XOR antara *state* sekarang *round key*.
3. *Final round*: proses untuk putaran terakhir:
   a. SubBytes
   b. ShiftRows
   c. AddRoundKey

6. Algoritma *Twofish*

Twofish merupakan algoritma yang beroperasi dalam mode *block*. Algoritma *Twofish* sendiri merupakan pengembangan dari algoritma *Blowfish*. Perancangan *Twofish* dilakukan dengan memperhatikan kriteria-kriteria yang diajukan *National Institute of Standards and Technology* (NIST) untuk kompetisi *Advanced Encryption Standard* (AES). Tujuan dari perancangan *Twofish* yang selaras dengan kriteria NIST untuk AES adalah sebagai berikut:

1. Merupakan *cipher* blok dengan kunci simetri dan blok sepanjang 128 bit.
2. Panjang kunci yang digunakan adalah 128 bit, 192 bit. Dan 256 bit.
3. Tidak mempunyai kunci lemah.
4. Efisiensi algoritma, baik pada Intel Pentium Pro dan perangkat lunak lainnya dan platform perangkat keras.
5. Rancangan yang *fleksibel*. Rancangan yang *fleksibel* ini dapat diartikan misalnya dapat menerima panjang kunci tambahan, dapat diterapkan pada platform dan aplikasi yang sangat variatif, serta cocok untuk *cipher* aliran, fungsi hash, dan MAC.

6. Rancangan yang sederhana agar memudahkan proses analisis dan implementasi algoritma.

2.3 Algoritma *Data Encryption Standard (DES)*

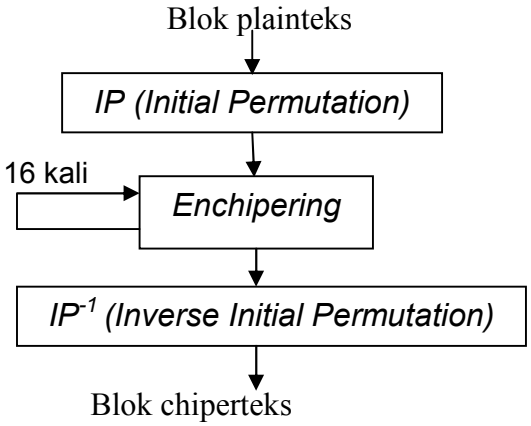
2.3.1 Sejarah *DES*

Pada sekitar akhir tahun 1960, IBM melakukan riset pada bidang kriptografi yang pada akhirnya disebut *Lucifer*. *Lucifer* dijual pada tahun 1971 pada sebuah perusahaan di london. *Lucifer* merupakan algoritma berjenis Block Cipher yang artinya bahwa input maupun output dari algoritma tersebut merupakan 1 blok yang terdiri dari banyak bit seperti 64 bit atau 128 bit.Lucifer beroperasi pada blok input 64 bit dan menggunakan key sepanjang 128 bit.

Lama-kelamaan *Lucifer* semakin dikembangkan agar bisa lebih kebal terhadap serangan analisis cypher tetapi panjang kuncinya dikurangi menjadi 56 bit dengan masksud supaya dapat masuk pada satu chip. Di tempat yang lain, biro standar amerika sedang mencari-cari sebuah algoritma enkripsi untuk dijadikan sebagai standar nasional.IBM mencoba mendaftarkan algoritmanya dan di tahun 1977 algoritma tersebut dijadikan sebagai *DES* (*Data Encryption Standard)*.

2.3.2 Proses Kerja Metode *DES*

DES termasuk ke dalam sistem kriptografi simetri dan tergolong jenis cipher blok. DES beroperasi pada ukuran blok 64 *bit*. DES mengenkripsikan 64 *bit* plainteks menjadi 64 *bit* chiperteks dengan menggunakan 56 *bit* kunci internal (*internal key*) atau upa kunci (*subkey*). Kunci internal dibangkitkan dari kunci eksternal (*external key*) dengan panjang kunci eksternal DES hanya 64 bit atau 8 karakter, itupun yang dipakai hanya 56 bit. Pada rancangan awal, panjang kunci yang diusulkan IBM adalah 128 bit, tetapi atas permintaan NSA, panjang kunci diperkecil menjadi 56 bit, maka DES memiliki kunci yang lemah dan desain struktur internal DES dimana bagian subsitusinya (S-box) masih dirahasiakan.



Gambar 2.7 Skema global algoritma DES (Munir, 2006)

Skema global dari algoritma DES adalah sebagai berikut :

1. Blok plainteks dipermutasi dengan matriks permutasi awal (*initial permutation* atau *IP*).
2. Hasil permutasi awal kemudian di *enchipering* sebanyak 16 kali putaran. Setiap putaran menggunakan kunci internal yang berbeda.
3. Hasil *enchipering* kemudian dipermutasi dengan matriks permutasi balikan *(invers initial permutation* atau IP^{-1}) menjadi bloks chiperteks.

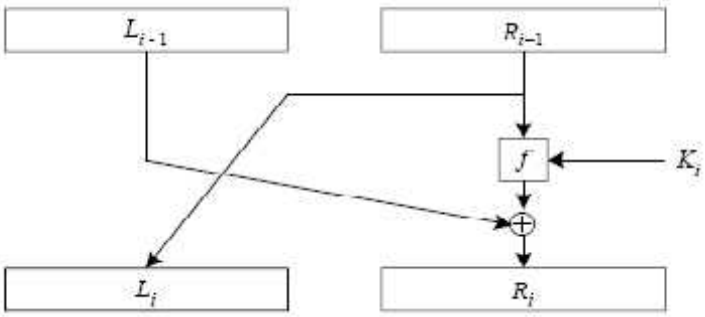
Di dalam proses *enciphering*, blok plainteks terbagi menjadi dua bagian, kiri (*L*) dan kanan (*R*), yang masing-masing panjangnya 32 bit. Kedua bagian ini masuk ke dalam 16 putaran DES. Pada setiap putaran *i*, blok *R* merupakan masukan untuk fungsi transformasi yang disebut *f*. Pada fungsi *f*, blok *R* dikombinasikan dengan kunci internal *Ki*. Keluaran dari fungsi *f* di-*XOR*kan dengan blok *L* untuk mendapatkan blok *R* yang baru. Sedangkan blok *L* yang baru langsung diambil dari blok *R* sebelumnya. Ini adalah satu putaran DES.

Secara matematis, satu putaran DES dinyatakan sebagai :

$Li = Ri - 1$

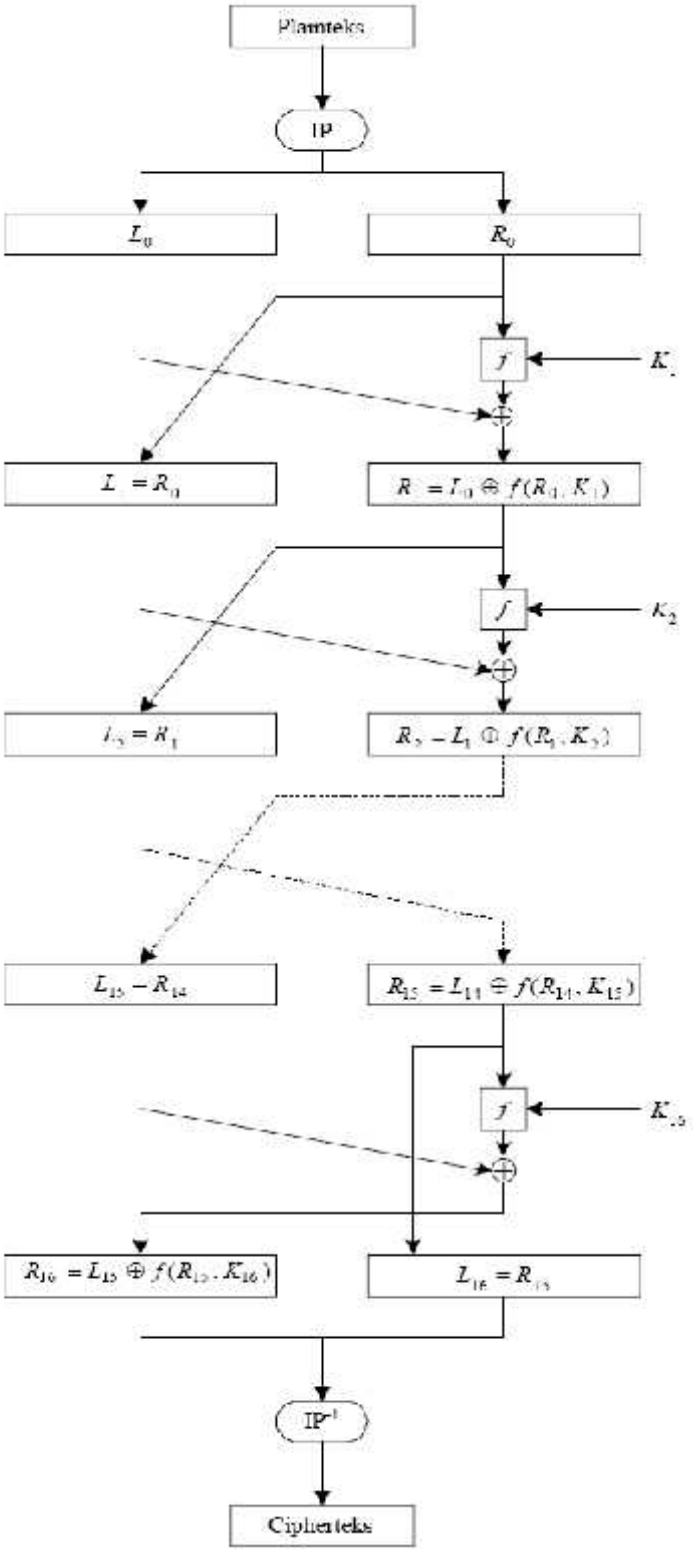
$Ri = Li - 1 \oplus f(Ri - 1, Ki)$

Satu putaran DES merupakan model jaringan *Feistel*



Gambar 2.8 Jaringan *Feistel* untuk satu putaran DES (Munir,2004)

Pada gambar diatas bahwa jika (L_{16}, R_{16}) merupakan keluaran dari putaran ke 16, maka (R_{16}, L_{16}) merupakan merupakan pra-cipherteks (*preciphertext*) dari *enciphering* ini. Cipherteks yang sebenarnya diperoleh dengan melakukan permutasi awal balikan, IP-1, terhadap blok pracipherteks.



Gambar 2.9 Algoritma Enkripsi dengan DES Permutasi Awal

Proses kerja kriptografi DES didahului dengan proses pembangkitan kunci yaitu kunci eksternal yang memiliki panjang 64 bit atau 8 karakter. Pembangkitan kunci dipermutasikan dengan berdasarkan matriks permutasi PC-1 (*Permutation Choice One*) sehingga menjadi 56 bit. Matriks-matriks dalam proses DES merupakan nilai konstanta yang telah ditetapkan NSA (*National Security Agency*)

Tabel 2.1. Matriks permutasi PC-1 (NSA (*National Security Agency*)) :

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

Hasil permutasi yang dibagi 2 blok proses pergeseran dengan aturan pada tabel 2.2 dan hasil dari pergeseran kedua blok tersebut digabungkan kembali dan dipermutasikan dengan menggunakan matriks PC-2 (*Permutation Choice Two*) sesuai dengan tabel 2.3.

Tabel 2.2 Aturan pergeseran yang diatur oleh putaran pada proses 16 iterasi (NSA (*National Security Agency*)).

Putaran (i)	Jumlah Pergeseran Bit
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2

14	2
15	2
16	1

Tabel 2.3 Matriks permutasi PC-2 (NSA (*National Security Agency*)) :

14	17	11	24	1	5	2	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

Algoritma DES memiliki proses selanjutnya yaitu enkripsi dan dekripsi. Enkripsi adalah proses dimana pesan yang disebut dengan plainteks di ubah menjadi chiperteks, dan dekripsi merupkan proses kebalikan dari enkripsi yaitu sirkulasi mengembalikan chiperteks menjadi plainteks kembali atau pesan awal. Proses enkripsi dan dekripsi dilakukan berdasarkan matrik *Initial Permutation* (IP) yang bertujuan untuk mengacak plainteks. Berikut matriks IP :

Tabel 2.4 Matriks Initial Pemutasi(NSA (*National Security Agency*)) :

58	50	34	26	18	10	2	60	52	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

Plainteks tersebut dibagi menjadi dua blok dan diproses putaran sebanyak 16 iterasi (Lampiran C). Setelah itu proses penggabungan kembali yang merupakan proses terkahir dengan permutasi yang menggunakan matriks permutasi IP^{-1}.

Tabel 2.5 Matriks inverse permutasi (NSA (*National Security Agency*)) :

40	8	48	16	56	24	64	32	39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30	37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28	35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26	33	1	41	9	49	17	57	25

2.3.3 Keamanan *DES*

Isu-isu yang menjadi controversial mengenai keamanan DES, yaitu (Munir,2006) :

1. Panjang kunci

Serangan yang paling praktis terhadap DES adalah *exhaustive key search*. Dengan panjang kunci 56 bit akan terdapat 2^{56} atau 72.057.594.037.927.936 kemungkinan kunci. Jika diasumsikan *exhaustive key search* dengan menggunakan prosesor paralel mencoba setengah dari jumlah kemungkinan kunci itu, maka dalam satu detik dapat dikerjakan satu juta serangan. Jadi seluruhnya diperlukan 1142 tahun untuk menemukan kunci yang benar.

2. Jumlah Putaran

Sebenarnya 8 putaran sudah cukup untuk membuat chiperteks sebagai fungsi acak dari setiap bit plainteks dan setiap bit chiperteks. Tetapi alasan yang mengharuskan 16 kali putaran (lampiran C) adalah karena sesuai dengan penelitian, DES dengan jumlah putaran yang kurang dari 16 kali putaran ternyata dapat dipecahkan dengan known-plaintext attack lebih mangkus daripada dengan brute force attack (John Wiley & Sons,1996).

3. Kotak-*S* (*S-box*)

Pengisian kotak-*S DES* (lampiran C) masih menjadi misteri tanpa ada alasan mengapa memilih konstanta-konstanta didalam kotak tersebut. Pengisian ini ditetapkan oleh *NSA (National Security Agency)*.

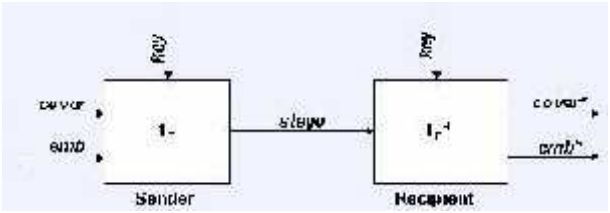
2.4 Steganografi

Steganografi merupakan ilmu dan seni menyembunyikan pesan rahasia (*hiding message*) ke dalam pesan lainnya sedemikian rupa sehingga keberadaan (eksistensi) pesan tidak terdeteksi oleh indra manusia (Munir, 2004). Kata steganografi (*steganography*) berasal dari bahasa yunani yaitu *steganos* yang artinya tersembunyi atau terselubung dan *graphein* atau *graptos* memiliki arti tulisan, maka dapat disimpulkan berarti "tulisan yang tersembunyi atau terselubung" (Sellars, 1996). Steganografi *digital* menggunakan media *digital*

sebagai penampung, misalnya citra, suara, teks, dan *video*. Steganografi konvensional berusaha merahasiakan komunikasi dengan cara menyembunyikan pesan ataupun mengklamufase pesan, maka sesungguhnya prinsip dasar dalam steganografi lebih dikonsentrasikan pada kerahsiaan komunikasinya bukan pada datanya (Johnson, 1995).

Tujuan dari steganografi adalah merahasiakan atau menyembunyikan keberadaan sebuah pesan tersembunyi atau sebuah informasi. Dalam prakteknya kebanyakan diselesaikan dengan membuat perubahan tipis terhadap data digital lain yang isinya tidak akan menarik perhatian dari penyerang potensial, sebagai contoh sebuah gambar yang terlihat tidak berbahaya. Perubahan ini bergantung pada kunci (sama pada cryptography) dan pesan untuk disembunyikan. Orang yang menerima gambar kemudian dapat menyimpulkan informasi terselubung dengan cara mengganti kunci yang benar ke dalam algoritma yang digunakan.

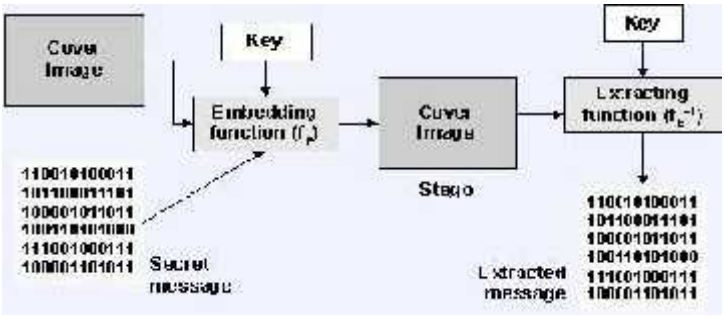
Secara garis besar sistem steganografi dapat dilihat pada gambar 2.11



Gambar 2.10 Sistem Steganografi (Henry, 2007)

Gambar 2.11 menunjukkan sebuah sistem steganografi umum dimana di bagian pengirim pesan (*sender*), dilakukkan proses *embedding* (f_E) pesan yang hendak dikirim secara rahasia (emb) ke dalam *data cover* sebagai tempat meyimpannya (*cover*), dengan menggunakan kunci tertentu (*key*), sehingga dihasilkan data dengan pesan tersembunyi di dalamnya (*stego*). Di bagian penerima pesan (*recipient*), dilakukkan proses *extracting* (f_E^{-1}) pada stego untuk memisahkan pesan rahasia (emb) dan data penyimpan (*cover*) tadi dengan menggunakan kunci yang sama seperti pada proses *embedding* tadi. Jadi hanya orang yang tahu kunci ini saja yang dapat mengekstrak pesan rahasia tadi.

Proses tadi dapat direpresentasikan secara lebih jelas pada gambar 2.10 di bawah:



Gambar 2.11 Diagram Sistem Steganografi (Henry,2007)

Penyembunyian data rahasia ke dalam media digital dapat mengubah kualitas media tersebut. Menurut Munir kriteria yang harus diperhatikan dalam menyembunyikan data diantaranya adalah: (Munir, 2004)

a. *Fidelity*, mutu citra penampung tidak jauh berubah. Setelah penambahan data rahasia, citra hasil steganografi masih terlihat dengan baik. Pengamat tidak mengetahui kalau di dalam citra tersebut terdapat data rahasia.
b. *Robustness*, data yang disembunyikan harus tahan terhadap manipulasi yang dulakukan pada citra penampung. Bila pada citra dilakukan operasi pengolahan citra, maka data yang disembunyikan tidak rusak.
c. *Recovery*. Data yang disembunyikan harus dapat diungkapkan kembali (*recovery*), karena tujuan steganografi adalah *data hiding*, maka sewaktu-waktu data rahasia di dalam citra penampung harus dapat diambil kembali untuk digunakan lebih lanjut.

Media yang telah disisipkan data atau pesan rahasia pada proses steganografi disebut *stegomessage*, sedangkan proses penyembunyian data kedalam media disebut penyisipan (*embedding*) dan proses sebaliknya disebut ekstraksi. Selain itu pada proses steganografi dilakukan dengan memasukkan atau menambahkan kunci, hal ini bertujuan untuk lebih meningkatkan keamanan.

2.4.1 Manfaat Steganografi

Steganografi dapat digunakan untuk menyembunyikan informasi rahasia, untuk melindunginya dari pencurian dan dari orang yang tidak berhak untuk mengetahuinya. Tetapi juga dapat digunakan untuk mencuri data yang disembunyikan pada data lain sehingga dapat dikirim ke pihak lain, yang tidak berhak, tanpa ada yang curiga. Steganografi juga dapat digunakan oleh para teroris untuk saling berkomunikasi satu dengan yang lain. Sehubungan dengan keamanan sistem informasi, tujuan steganografi untuk menyembunyikan pesan rahasia dan dapat dianggap sebagai pelengkap dari kriptografi. Steganografi lebih cocok digunakan bersamaan dengan metode lain tersebut untuk menciptakan keamanan yang ganda.

2.4.2 Metode Steganografi

Steganografi dapat diterapkan pada data digital, yaitu teks, citra, suara, dan *video*.Ada empat jenis metode Steganography, yaitu :

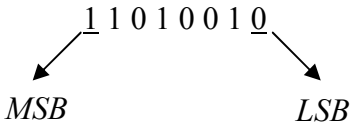
1. Least Significant Bit (LSB).
2. Algorithms and Transformation.
3. Redundant Pattern Encoding.
4. Spread Spectrum method.

2.4.2.1 Least Significant Bit (LSB)

Cara paling umum untuk menyembunyikan pesan adalah dengan memanfaatkan *Least Significant Bit* (LSB). Kemudahan implementasinya membuat metode ini tetap digunakan sampai sekarang. Metode ini membutuhkan syarat, yaitu jika dilakukan kompresi pada *stego*, harus digunakan *format lossless compression*, karena metode ini menggunakan *bit-bit* setiap *piksel* pada *image*.

Penyembunyian data dilakukan dengan mengganti beberapa *bit data* di dalam *biner* citra digital dengan *bit-bit data* rahasia. Pada susunan *bit* di dalam sebuah *byte* (1 *byte* = 8 *bit*), ada *bit* yang paling berarti (*Most Significant Bit*) dan *bit* yang paling kurang berarti (*Least Significant Bit*).

Contoh:



Bit yang cocok untuk diganti adalah *bit LSB*, karena perubahan tersebut hanya mengubah nilai *byte* satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya. Misalkan *byte* tersebut menyatakan warna merah, maka perubahan satu *bit LSB* tidak mengubah warna merah tersebut secara berarti. Lagi pula, indra penglihatan manusia tidak dapat membedakan perubahan yang kecil. Jika digunakan *image* 24 *bit color* sebagai *cover*, sebuah *bit* dari masing-masing komponen *Red, Green,* dan *Blue* (RGB), yang masing-masing disusun oleh bilangan 8 bit dari 8 bit (byte) dari 0 sampai 255 atau dengan format biner 00000000 sampai 11111111.

Misalnya, di bawah ini terdapat 3 *piksel* dari *image* 24 bit color :

(00100111 11101001 11001000)

(00100111 11001000 11101001)

(11001000 00100111 11101001)

Jika diinginkan untuk menyembunyikan karakter A (10000001) dihasilkan :

(00100111 1110100**0** 11001000)

(0010011**0** 11001000 1110100**0**)

(11001000 00100111 11101001)

dapat dilihat bahwa hanya 3 *bit* saja yang perlu diubah untuk menyembunyikan karakter A ini. Perubahan pada LSB ini akan terlalu kecil untuk terdeteksi oleh mata manusia sehingga pesan dapat disembunyikan secara efektif. Jika digunakan *image* 8 bit *color* sebagai *cover*, hanya 1 *bit* saja dari setiap *piksel* warna yang dapat dimodifikasi sehingga pemilihan *image* harus dilakukan dengan sangat hati-hati, karena perubahan LSB dapat menyebabkan terjadinya perubahan warna yang ditampilkan pada citra. Akan lebih baik jika image berupa *image grayscale* karena perubahan warnanya akan lebih sulit dideteksi oleh mata manusia. Proses ekstraksi pesan dapat dengan mudah dilakukan dengan mengekstrak LSB dari

masing-masing piksel pada *stego* secara berurutan dan menuliskannya ke *output file* yang akan berisi pesan tersebut. Kekurangan dari metode modifikasi LSB ini adalah bahwa metode ini membutuhkan tempat penyimpanan yang relatif besar. Kekurangan lain adalah bahwa *stego* yang dihasilkan tidak dapat dikompress dengan *format lossy compression*.

Secara umum dibawah ini akan dijelaskan langkah-langkah algoritma *Least Significant Bit (LSB*) teks pada gambar adalah (Munir, 2004):

1. Pilih citra *bitmap* sebagai citra penampung dengan *format* 24 *bit* yang sudah tersusun atas komponen *RGB*.
2. Konversikan citra penampung ke bentuk *biner*.
3. Pilih teks dan karakter yang akan disembunyikan.
4. Konversikan teks dan karakter ke bentuk *biner*
5. Setiap *byte* di dalam citra *bitmap* diganti (subtitusi) satu *bit LSB*-nya (*bit* terakhir yang paling kurang berarti) dengan *bit data* teks dan karakter yang akan disembunyikan.
6. Contoh penyisipan:

Segmen *pixel-pixel* sebelum penambahan *bit-bit* rahasia (yang telah dikonversikan ke sistem *biner*).

00110011101000101110001001101111

Data rahasia (yang telah dikonversikan ke sistem *biner*) adalah **0111**.

Setiap *bit* dari data rahasia menggantikan posisi *LSB* dari segmen data citra menjadi:

00110010101000111110001101101111

2.5 Bilangan Biner

Sistem bilangan biner adalah susunan bilangan yang mempunyai basis 2 sebab sistem bilangan ini menggunakan dua nilai koefisien yang mungkin yaitu **0** dan **1**. berikut erupakan tabel yang menggambarkan basis bilangan biner dan bilangan desimal.

Tabel 2.6 Basis Bilangan Desimal dan Biner

Bilangan Desimal	Bilangan Biner
0	**0000**
1	**0001**
2	**0010**
3	**0011**
4	**0100**
5	**0101**
6	**0110**
7	**0111**
8	**1000**
9	**1001**
A	**1010**
B	**1011**
C	**1100**
D	**1101**
E	**1110**
F	**1111**

2.6 Citra Digital

Citra digital merupakan suatu gambar yang tersusun dari *pixel*, dimana tiap *pixel* merepresentasikan warna (tingkat keabuan untuk gambar hitam putih) pada suatu titik di gambar. Gambar digital merupakan dokumen berbentuk *file* yang dihasilkan melalui perangkat elektronik atau media digital. Gambar digital berupa sekumpulan titik yang disusun dalam bentuk matriks, dan nilainya menyatakan suatu derajat kecerahan (derajat keabuan/*gray-scale*). Citra digital mengandung sejumlah elemen dasar. Elemen-elemen dasar tersebut dimanipulasi dalam pengolahan citra dan dieksploitasi lebih lanjut dalam *computer vison*. Elemen-elemen dasar yang penting pada citra digital di antaranya adalah (Munir,2004) :

1. Kecerahan (*brightness*)

2. Kontras (*contrast*)
3. Kontur (*contour*)
4. Warna (*color*)
5. Bentuk (*shape*)
6. Tekstur (*texture*)

Citra digital disimpan juga secara khusus di dalam file 24-bit atau 8-bit. Gambar 24-bit menyediakan lebih banyak ruang untuk menyembunyikan informasi, Semua variasi warna untuk pixel yang diperoleh dari tiga warna dasar yaitu merah, hijau dan biru (RGB). Setiap warna dasar direpresentasikan dengan 1 byte. gambar 24-bit menggunakan 3 byte per pixel untuk merepresentasikan suatu nilai warna. 3 byte ini dapat direpresentasikan sebagai nilai hexa-desimal, desimal, dan biner. Definisi latar belakang putih adalah analog dengan definisi warna dari pixel tunggal dalam suatu gambar. Pixel merepresentasikan kontribusi pada ukuran file.

Derajat keabuan 8 bit menyatakan 256 derajat kecerahan. Pada gambar berwarna nilai setiap titiknya adalah nilai derajat keabuan pada setiap kompoen warna RGB. Bila masing-masing komponen R,G dan B mempunyai 8 bit, maka satu titik dinyatakan dengan (8+8+8)=24 bit atau 2^{24} derajat keabuan.

Secara garis besar, gambar dapat dibagi menjadi menjadi beberapa tipe atau format, diataranya:

1. BMP *(BitMap Graphics)*.

   Kriteria yang paling penting dari citra ini adalah kedalaman warna yaitu berapa banyak bit per pixel yang didefinisikan dari sebuah warna (Rinaldi Munir, 2005). Bitmap dengan mengikuti kriteria tadi maka dapat dilihat:

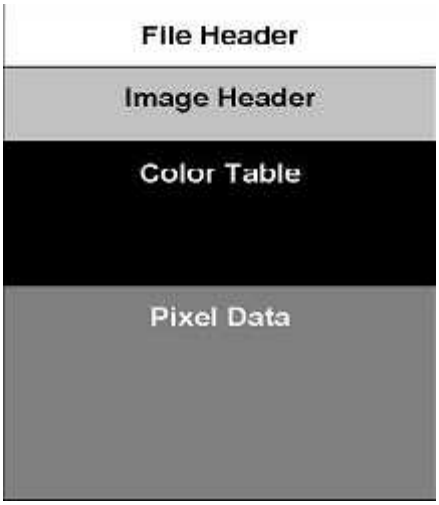
   a. 8 bit = 256 warna (256 gray scales).

   b. 24 bit = 16.777.216 warna

   Tipe file BMP umum digunakan pada sistem operasi *Windows*. Kelebihan file BMP adalah dapat dibuka oleh hampir semua program pengolah gambar. Baik file BMP yang terkompresi maupun tidak terkompresi, file BMP memiliki ukuran yang jauh lebih besar dari pada tipe-tipe yang lain.

Struktur file BMP terdiri dari 4 bagian, yaitu: *File Header, Image Header, Color Table* dan *Data Pixel*. Header file BMP (*File Header* + *Image Header* + *Color Table*) biasanya sebesar 54 byte.

Struktur file BMP dapat dilihat dari gambar berikut:



Gambar 2.12 Struktur *File* BMP

2. JPG / JPEG (*Joint Photographic Expert Group*)

   Format ini didesain untuk gambar-gambar dengan keadalaman warna 24-bit. *File* JPG menggunakan teknik kompresi yang menyebabkan kualitas gambar turun (*lossy compression*). Setiap kali menyimpan ke tipe JPG dari tipe lain, ukuran gambar biasanya mengecil, dan kualitasnya turun dan tidak dapat dikembalikan lagi. Ukuran *file* BMP dapat turun menjadi sepersepuluh setelah dikonversi menjadi JPG. Meskipun dengan penurunan kualitas gambar, pada gambar-gambar tertentu (misalnya pemandangan), penurunan kualitas gambar hampir tidak terlihat mata.

3. GIF (*Graphics Interchange Format*)

   File GIF memungkinkan penambahan warna transparan dan dapat digunakan untuk membuat animasi sederhana, tetapi saat ini standar GIF hanya maksimal 256 warna saja. *File* ini menggunakan kompresi yang tidak menghilangkan data *(lossles compression)* tetapi penurunan jumlah warna menjadi 256 sering membuat gambar yang kaya warna seperti pemandangan menjadi tidak realistis.

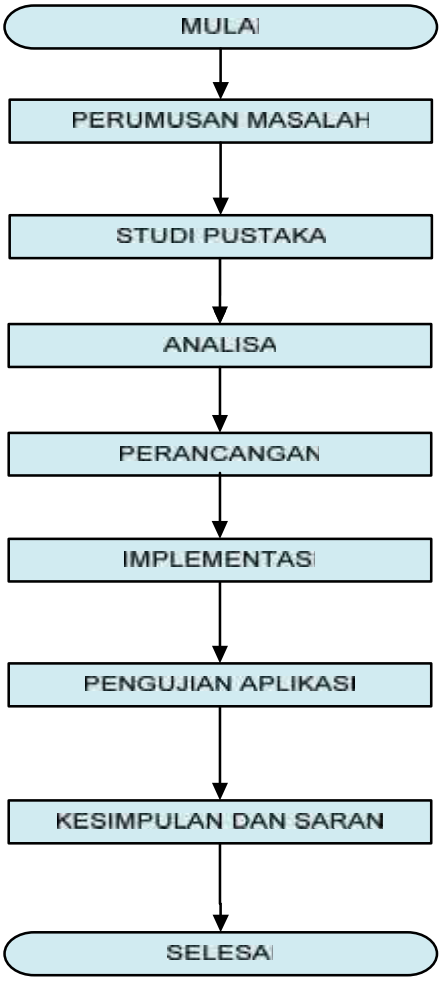
4. PNG (*Portable Network Graphics*).

   Tipe file PNG merupakan solusi kompresi yang *powerfull* dengan warna yang lebih banyak (24 bit RGB + *alpha*). Berbeda dengan JPG yang menggunakan teknik kompresi yang menghilangkan data, *file* PNG menggunakan kompresi yang tidak menghilangkan data (*lossles compression*). Kelebihan *file* PNG adalah adanya warna transparan dan alpha. Warna alpha memungkinkan sebuah gambar transparan, tetapi gambar tersebut masih dapat dilihat mata seperti samar-samar atau bening. File PNG dapat diatur jumlah warnanya 64 bit (*true color* +*alpha*) sampai *indexed color* 1 bit. Dengan jumlah warna yang sama, kompresi file PNG lebih baik daripada GIF, tetapi memiliki ukuran *file* yang lebih besar daripada JPG.

BAB III

METODOLOGI PENELITIAN

Metodologi penelitian merupakan sistematika tahapan penelitian yang menguraikan seluruh kegiatan yang dilaksanakan selama kegiatan penelitian berlangsung selama pembuatan tugas akhir. Berikut langkah-langkah yang dilalui dalam pelaksanaan penelitian :



Gambar 3.1 Diagram Alir Metodologi Penelitian

Berdasarkan gambar 3.1 metodologi penelitian dalam pengerjaan Tugas Akhir meliputi beberapa tahapan, yaitu :

3.1 Perumusan Masalah

Tahap ini merupakan tahap mengidentifikasi masalah dengan memanfaatkan informasi-informasi yang didapat dari penelitian pendahuluan yang telah dilaksanakan. Pada tugas akhir ini masalah yang akan diidentifikasi adalah melakukan studi analisis pengamanan pesan yang berupa *text* dengan menggunakan metode algoritma *Data Encryption Standard (DES)* dan *Least Significant Bit Insertion (LSB)* diimplementasikan dalam penyandian citra digital berformat *bitmap*.

3.2 Studi Pustaka

Tahap pengumpulan data yang dilakukan untuk mendapatkan teori-teori lanjutan yang dibutuhkan, yaitu dengan melakukan Studi Pustaka (*Library Research*). Studi pustaka ini bertujuan mendapatkan dasar-dasar pengetahuan yang akan diterapkan dalam penelitian. Teori-teori diperoleh dengan cara mempelajari buku-buku dan literatur yang berkaitan dengan penelitian algoritma *Data Encryption Standard* (*DES)* dan *Least Significant Bit (LSB)*, dan penyandian pesan dalam citra.

3.3 Analisa

Menganalisa untuk merancang suatu aplikasi pengamanan pesan yang menggunakan citra *bitmap* 24 bit sebagai media objeknya dengan metode algoritma *Data Encryption Standards* dan *Least Significant Bit*. Analisa perangkat lunak dalam membangun aplikasi ini meliputi analisa masalah, analisa metode, analisa kebutuhan sistem dari aplikasi yang akan dibuat, sehingga aplikasi yang dibangun dan diimplementasikan sesuai dengan maksud dan tujuan yang ingin dicapai dalam penelitian ini.

3.3.1 Gambaran Umum Aplikasi

Dalam kriptografi, pesan yang tersimpan disandikan dalam bentuk yang tidak dapat dipahami agar makna pesan tidak dimengerti oleh pihak lain dengan aspek keamanan seperti kerahasiaan, integritas data, serta autentikasi. Untuk

memperkuat keamanan data, dapat dikombinasikan dengan steganografi, yang merupakan penyisipan pesan. Berdasarkan penelitian - penelitian sebelumnya, steganografi juga disebut sebagai pelengkap dari kriptografi. Pada penelitian ini, membahas pengamanan pesan yang berupa teks dalam gambar bitmap yang menjadi media objek dari penyimpanan.

3.3.2 Gambaran Umum Analisis Metode Data Encryption Standard (DES) dan Least Significant Bit (LSB)

Metode yang digunakan dalam eksistensi pengamanan pesan yang berupa teks menggunakan algoritma *Data Encryptoin Standards* dan *Least Significant Bit* yang merupakan implementasi dari kriptografi dan steganografi.

Dalam analisa metode ini akan menjelaskan langkah – langkah proses kerja dari aplikasi penelitian ini secara rinci.

3.4 Perancangan Aplikasi

Setelah melakukan analisa, kemudian dilanjutkan dengan perancangan sistem berdasarkan analisa permasalahan yang telah dilakukan sebelumnya.

3.4.1 Perancangan Fungsional

Rancangan fungsional meliputi analisa sistem yang akan dibangun dengan perancangan diagram alir (*flowchart*), *context diagram*, dan *data flow diagram* (DFD).

3.4.2 Perancangan Antar Muka (*Interface*) Aplikasi

Untuk mempermudah komunikasi antara sistem dengan pengguna, maka perlu dirancang antar muka (*interface*). Dalam perancangan *interface* hal terpenting yang ditekankan adalah bagaimana menciptakan tampilan yang baik dah mudah dimengerti oleh pengguna.

3.4.3 Perancangan Struktur Menu

Rancangan struktur menu diperlukan untuk memberikan gambaran terhadap menu-menu atau *fitur* pada sistem yang akan dibangun.

3.5 Implementasi

Pada tahap implementasi merupakan tahap penerjemahan hasil analisa kedalam bentuk *coding* sesuai dengan hasil perancangan yang telah dibuat dan pengujian (*testing*) dimana aplikasi siap dioperasikan pada keadaan yang sebenarnya, sehingga dapat diketahui apakah aplikasi yang dibuat telah mencapai tujuan yang diinginkan. Implementasi keamanan data menggunakan teknik dan seni kriptografi dan steganografi teks pada citra format *bitmap* dengan metode *Data Encryption Standards* dan *Least Significant Bit Insertion* menggunakan bahasa pemrograman *Microsoft Visual Basic 6.0.*

3.6 Pengujian

Tahapan pengujian dilakukan bila tahapan implementasi aplikasi pengamanan pesan menggunakan algoritma *Data Encryption Standards* dan *Least Significant Bit* pada citra digital telah dilakukan. Pada tahap ini dilakukan pengujian secara fungsional. Pengujian fungsional merupakan pengujian yang berhubungan dengan kinerja sistem secara intern, berupa respon sistem terhadap *user*, uji fungsi atau menu yang terdapat pada sistem, dan uji kerja sistem. Pengujian ini dilakukakan menggunakan *black box* dan berdasarkan *fidelity* atau kapasitas gambar citra penampung. Pengujian dari aplikasi ini juga dilakukan dengan *steganalisis tools* yaitu dengan menggunakan aplikasi *StegSpy V 2.1* produksi *Spy Hunter* pada website www.spy-hunter.com untuk mendeteksi keberadaan pesan rahasia dalam gambar hasil steganografi.

3.7 Kesimpulan dan Saran

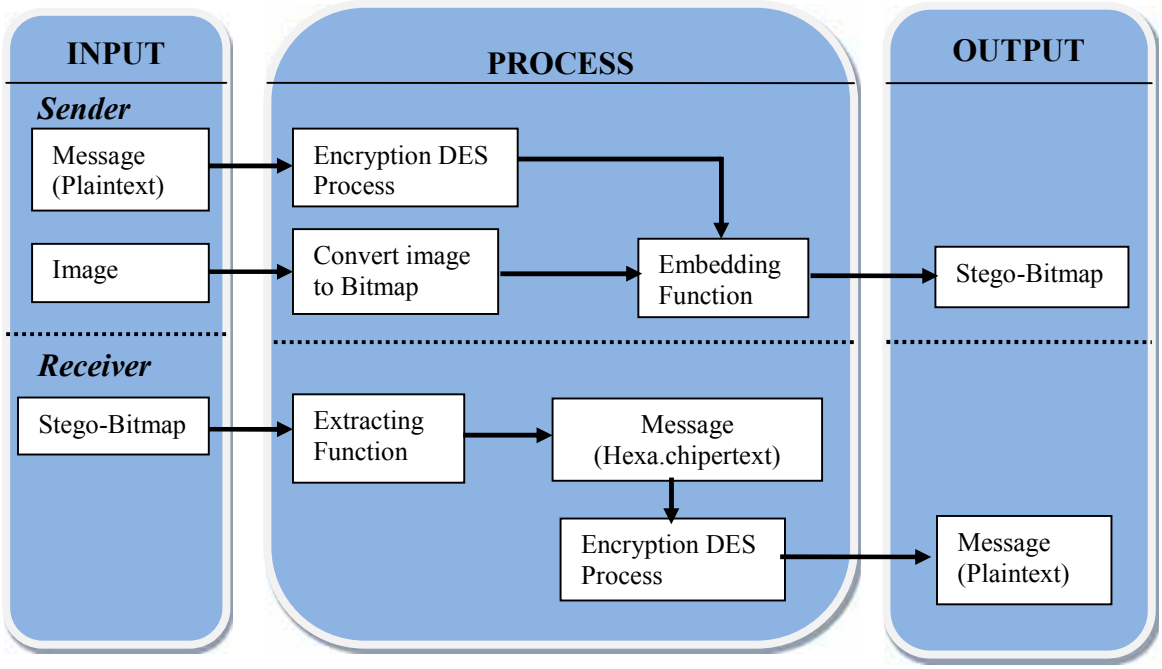
Pada bagian ini, berdasarkan hasil pengujian dihasilkan kesimpulan mengenai hasil evaluasi dari seluruh kegiatan yang dilakukan dalam melakukan penelitian terhadap implementasi aplikasi pengamanan pesan menggunakan algoritma *Data Encryption Standards* dan *Least Significant Bit* pada citra digital, serta saran-saran yang diperlukan untuk pengembangan dan pengelolaan sistem lebih lanjut.

BAB IV

ANALISA DAN PERANCANGAN

4.1 Gambaran Umum Aplikasi

Konsep implementasi aplikasi pengamanan pesan yang menerapkan metode kriptografi dan steganografi pada dasarnya memiliki proses mengubah pesan berupa teks ke dalam bentuk yang tidak dimengerti menggunakan algoritma *Data Encryption Standard* (DES) dimana *sender* (pengirim) melakukan proses enkripsi berupa *input*an teks dan penyisipan (*embedding function*) yang merupakan proses steganografi metode *Least Significant Bit* (LSB). Pada *receiver* (penerima) merupakan proses hasil steganografi, pesan yang telah disisipkan dimulai dengan pengambilan pesan (*extracting function*) dari gambar sehingga didapatkan pesan untuk didekripsi menjadi pesan awal. Yang berperan sebagai objek media penyimpanan pada teknik pengamanan pesan ini adalah citra digital berformat *bitmap* yang berfungsi mengklamufasekan eksistensi pesan. Untuk lebih jelasnya dapat dilihat pada gambar dibawah ini :



Gambar 4.1 Gambaran Umum Aplikasi

Implementasi aplikasi gabungan kriptografi dan steganografi yang akan dibangun mempunyai *input*an-*input*an kebutuhan data pada umumnya, yaitu :

1. Data Teks

   Data teks yang berisi huruf, angka, maupun karakter teks *input*-an keyboard yang akan diproses enkripsi dan dekripsi terlebih dahulu akan dikonversikan kedalam bentuk heksadesimal sesuai dengan kode ASCII. *Hexaplaintext* tersebut konversikan ke biner untuk dapat melanjutkan ke proses enkripsi DES yang nantinya menghasilkan chiperteks berupa heksadesimal, kemudian disisipkan ke dalam gambar.

2. Media Objek Pengamanan

   Media objek pengamanan, disebut juga media penampung, yaitu media yang berupa citra digital dengan *form*at *bitmap* 24 bit sebagai penampung yang akan disisipkan teks.

3. *Password* (Kata Kunci)

   Kata kunci merupakan hak akses antara pengirim dan penerima pesan yang digunakan untuk keamanan *bitmap-stegano* dan dibutuhkan untuk membuka hasil steganografi.

4.1.1 Gambaran Umum Analisis Metode Data Encryption Standard (DES) dan Least Significant Bit (LSB)

Berdasarkan pada batasan masalah telah dipaparkan bahwa pesan hanya berupa karakter teks. Pada penelitian ini membangun aplikasi metode yang dikombinasikan antara *crypto-stego* yang bertujuan untuk meningkatkan keamanan yang berlapis. Metode DES merupakan teknik kriptografi dimana pesan teks yang disebut dengan *plaintext* diubah menjadi *chipertext*, secara global dipermutasikan dengan matriks *initial permutation* dan di *enchipering* sebanyak 16 iterasi, kemudian hasil *enchipering* dipermutasikan dengan matriks *inverse initial permutation* sehingga menghasilkan *chiperteks.* Pada *stego-image* metode LSB, teks hasil enkripsi dalam biner disisipkan dalam setiap bit-bit terakhir (bit

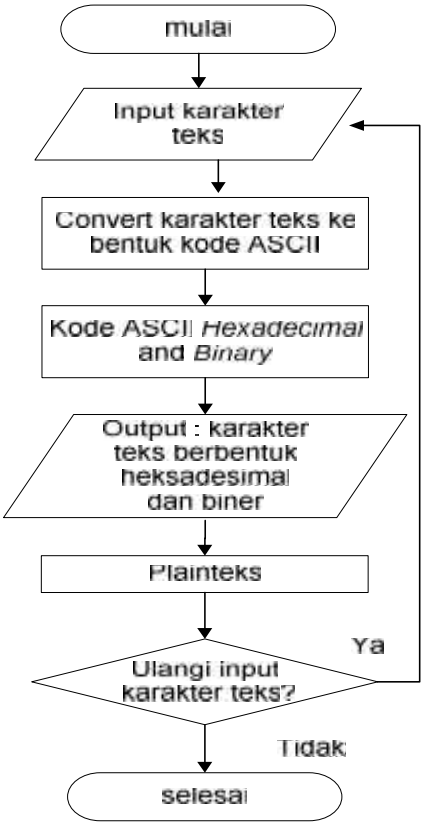
ke-8) bitmap, sehingga gambar bitmap hasil steganografi tidak mengalami perubahan.

Spesifikasi pemilihan media pengamanan pesan berupa gambar *form*at file BMP merupakan *form*at file standar sistem operasi yang mendukung resolusi warna dari Monochrom hingga True Color (16,7 juta warna). *Form*at file BMP 24 bit menggunakan model warna RGB. Pada model warna RGB, warna yang ditampilkan di layar monitor disusun oleh tiga buah warna primer, yaitu *Red* (merah), *Green* (hijau), *Blue* (Biru). Pada model warna RGB setiap titik pada layar monitor berisi angka yang menunjukkan intensitas kombinasi warna yaitu yang menentukan proporsi warna merah, hijau, biru yang dapat dipilih untuk mengisi warna pada sebuah pixel adalah 256 x 256 x 256 = 16,7 juta warna. Bitmap umum digunakan pada system operasi *windows*, yang merupakan keunggulan dasar karena dapat dibuka oleh hampir semua program pengolah citra grafis dan memiliki ukuran yang sangat lebih besar dari tipe-tipe yang lain.

4.1.2 Proses Konversi Karakter Teks Pada Proses DES

Teks yang menjadi *input*an dalam pesan rahasia akan dikonversikan ke dalam heksadesimal berdasarkan kode ASCII (lampiran B). Kemudian dilakukan proses terhadap tiap-tiap kode ASCII yang dihasilkan sehingga dapat dikonversikan lagi dalam bentuk biner.

Proses konversi karakter teks ke bentuk heksa dan biner melalui beberapa tahap yang dapat dilihat pada gambar berikut :

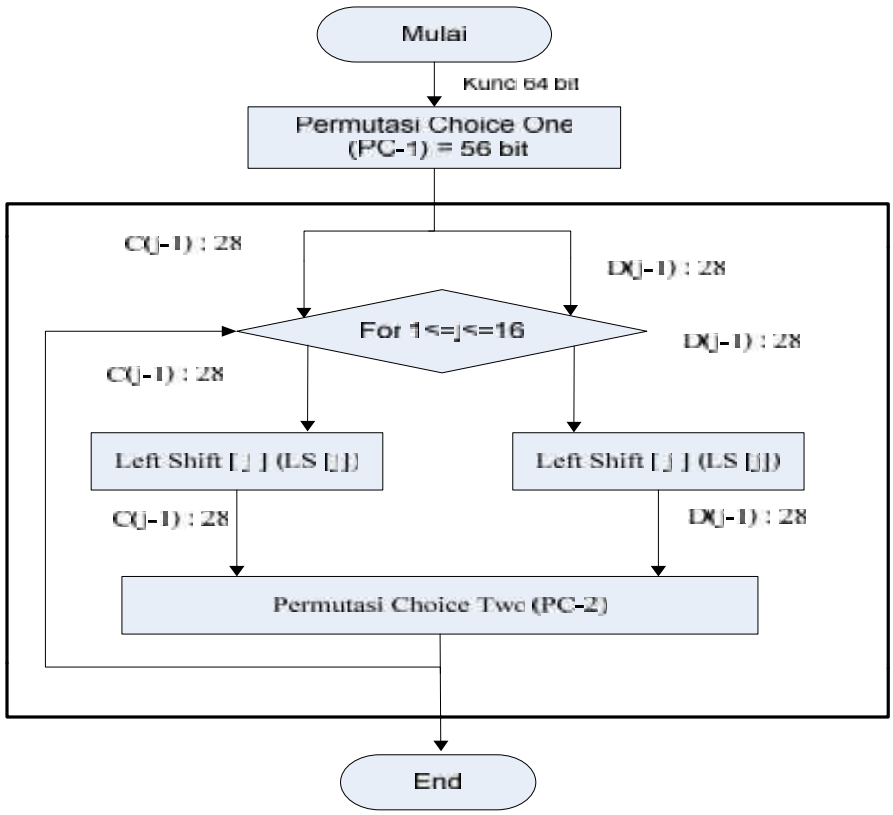


Gambar 4.2. *Flowchart* Konversi Karakter Teks Ke Bentuk Heksadesimal dan Biner

4.1.3 Proses Kriptografi *Data Encryption Standard* (DES)

4.1.3.1 Pembangkitan Kunci DES

Pada proses DES dilakukan proses pembangkitan kunci terlebih dahulu untuk membangkitkan kunci pada algoritma DES. *Flowchart* pembangkitan kunci pada algoritma DES :



Gambar 4.3 *Flowchart* pembangkitan kunci algoritma DES

Langkah-langkah dalam pembangkitan kunci adalah sebagai berikut (Berdasarkan Bab II Landasan Teori) :

a. Masukkan kunci eksternal, yang panjangnya 64 bit atau 8 karakter.
b. Kunci eksternal 64 bit tersebut dipermutasikan dengan matriks permutasi PC-1. Sehingga panjang kunci yang tadinya 64 bit menjadi 56 bit.
c. Hasil permutasi yang panjangnya 56 bit tersebut kemudian dibagi menjadi 2 blok, yaitu Cj dan Dj. Cj merupakan kumpulan dari bit pertama sampai bit 28 dan Dj merupakan kumpulan bit 29 sampai bit 56.
d. Kemudian pada kedua blok dilakukan pergeseran dengan aturan pada tabel 2.2 (Bab II).
e. Setelah mengalami proses penggeseran, kedua blok digabungkan kembali dan akan dipermutasikan dengan matriks permutasi PC-2.

f. Hasil permutasi dengan matriks permutasi PC-2 merupakan kunci internal yang akan digunakan untuk setiap putaran sebanyak 16 iterasi pada proses enkripsi dan dekripsi algoritma DES.

4.1.3.2 Enkripsi Pesan

Pesan yang ingin disampaikan merupakan dat rahasia yang diubah kedalam bentuk yang tidak dimengerti agar tidak dapat diketahui isi pesan tersebut. Pada penelitian ini metode yang digunakan adalah kritografi *Data Encryption Standard* yang memiliki aturan dan ketentuan tertentu baik dari pembangkitan kunci maupun dari matriks permutasi. Proses enkrispi DES adalah sebagai berikut :

a. *Input*kan teks yang akan dienkripsi (plainteks).
b. Teks tersebut akan diubah menjadi bilangan biner dengan mengacu kepada hexadesimal sesuai nilai indeks pada tabel *ASCII.*
c. Teks yang telah diubah menjadi bilangan biner tersebut dalam hal enkripsi dibagi menjadi 64 bit.
d. Teks yang telah dibagi tadi kemudian di permutasi dengan matriks permutasi awal (*Initial Permutation*).
e. Teks yang telah diacak tersebut kemudian dibagi menjadi dua blok, masing-masingnya adalah 32 bit. Kedua blok yang dilambangkan dengan L_0 dan R_0.
f. Proses selanjutnya yaitu melakukan putaran proses sebanyak 16 kali (Lampiran C). Proses yang dilakukan dalam setiap putaran adalah :

   $L_i = R_{i-1}$

   $R_i = L_{i-1} \Theta K_1$

g. Kemudian kedua blok tersebut digabungkan kembali.
h. Setelah penggabungan kedua blok, maka proses terakhir adalah melakukan permutasi dengan menggunakan matriks permutasi IP^{-1}.

4.1.3.3 Dekripsi Pesan

Proses dekripsi merupakan proses kebalikan dari proses enkripsi yang bertujuan untuk membalikkan data kembali menjadi in*form*asi semula (plainteks)

yang dapat digunakan oleh pengguna. Berikut adalah langkah-langkah proses dekripsi :

a. Yang menjadi *input* adalah chiperteks hasil enkripsi.
b. Kemudian pada chiperteks tersebut dilakukan proses invers permutasi dengan menggunakan matriks permutasi IP^{-1}.
c. Chiperteks dibagi menjadi dua blok seperti pada proses enkripsi.
d. Kemudian dilakukan perputaran sebanyak 16 kali.
e. Kedua blok tersebut digabungkan kembalidan dilakukan invers permutasi dengan menggunakan matriks permutasi IP.

4.1.4 Proses Steganografi Least Significant Bit (LSB)

Selanjutnya pada proses penyisipan pesan (*embedding function*) dan pengambilan pesan (*extracting function*) akan diproses menggunakan metode *Least Significant Bit* dan sebagai media penyimpanannya adalah bitmap. Didalam proses in terdapat dua file biner dimana file yang pertama adalah file biner hasil dari konvers bitmap kedalam biner dan file yang kedua adalah file biner kata teks yang telah melalui poses enkripsi yang akan disisipkan dikenversi kedalam file biner.

Konsep dari penyisipan ini adalah bahwa setiap delapan bit file pertama akan disisipkan satu bit file yang berasal dari file kedua yang hanya menggantikan bit-bit terakihir, karena metode ini menggunakan bit-bit setiap *pixel* pada *image*. Berikut ini gambaran dari proses tersebut :

File Pertama (contoh simulasi file biner hasil konvers dari bitmap ke biner)

```
11000011  11010011  11000011  10100110  00011101  01111000  01110111
01000001  10111011  01001110  11001100  00111010  01100000  11101101
10100111  00001100  00011100  11111010  11110000  11100001  10110011
11001100  00011011  01110000  11101110  11000011  10011111  00101110
11011100  10111011  10111010  01000001  11001111  10100111  00101100
00111101  00110010  11100111  11010011  11001110  01100110  11111110
11011001   01100000 10110011 10111111  10101111   00101000 00101011
```

11000011 11000011 00011101 01000001 01100000 11101101 11010011
11110000 10110011 11111010 11100001 10110011 10111111 10101111
00101000

File Kedua (contoh simulasi file biner hasil konvers file teks yang akan disisipkan ke file biner)

Chipertext hexa : 56 F1 D5 C8 52 AF 81 3F

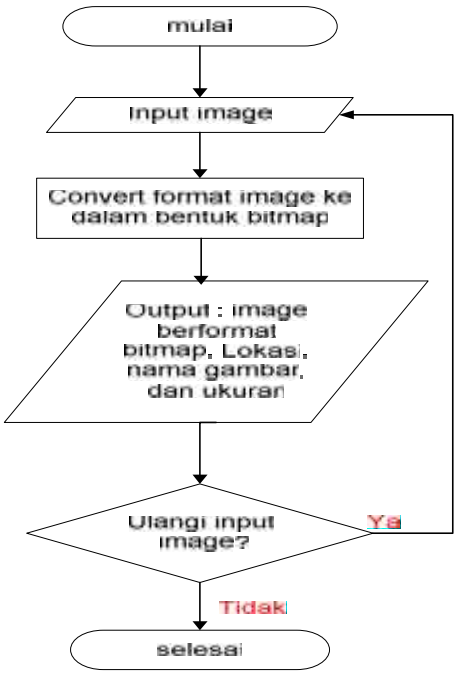
Chipertext biner : 01010110 11110001 11010101 11001000 01010010 10101111 10000001 00111111

File hasil penyisipan (biner yang ber garis bawah adalah hasil biner yang disipkan dari file kedua)

11000010 11010011 11000010 10100111 00011100 01111001 01110111
01000000 10111011 01001111 11001101 00111011 01100000 11101100
10100110 00001101 00011101 11111011 11100000 11000001 10110010
11001101 00011010 01110001 11101111 11000011 10011110 00101110
11011101 10111010 10111010 01000000 11001110 10100111 00101100
00111101 00110010 11100110 11010011 11001110 01100111 11111110
11011001 01100000 10110011 10111111 10101111 00101001 00101011
11000010 11000010 00011100 01000000 11100000 10110010 11010011
11100000 10110010 11111011 11100001 10110011 10111111 10101111
00101001

4.1.5 Proses Pemilihan Citra Penampung Dikonversi Dalam *Form*at Bitmap

Berdasarkan batasan masalah bahwa media penyimpanan hasil steganografi adalah citra digital ber*form*at bitmap. Proses ini merupakan memanipulasikan pesan chiperteks heksadesimal hasil enkripsi yang disisipkan dalam bitmap. Citra digital yang di*input* dalam berbagai *form*at yang nantinya pada proses steganografi akan dikonversikan dalam bentuk bitmap. Alur diagramnya dapat dilihat pada gambar 4.4 :



Gambar 4.4 *Flowchart* Proses Pemilihan Citra Digital**.**

4.2 Perancangan Aplikasi

Perancangan perangkat lunak dalam membangun aplikasi penggabungan kriptografi dan steganografi untuk penyandian citra menggunakan algoritma *Data Encryption Standard* dan *Least Significant Bit* meliputi perancangan procedural dan perancangan antar muka proses enkripsi dan dekripsi serta proses penyisipan pada citra digital *bitmap* dan pengambilan pesan.

4.2.1 Perancangan Fungsional

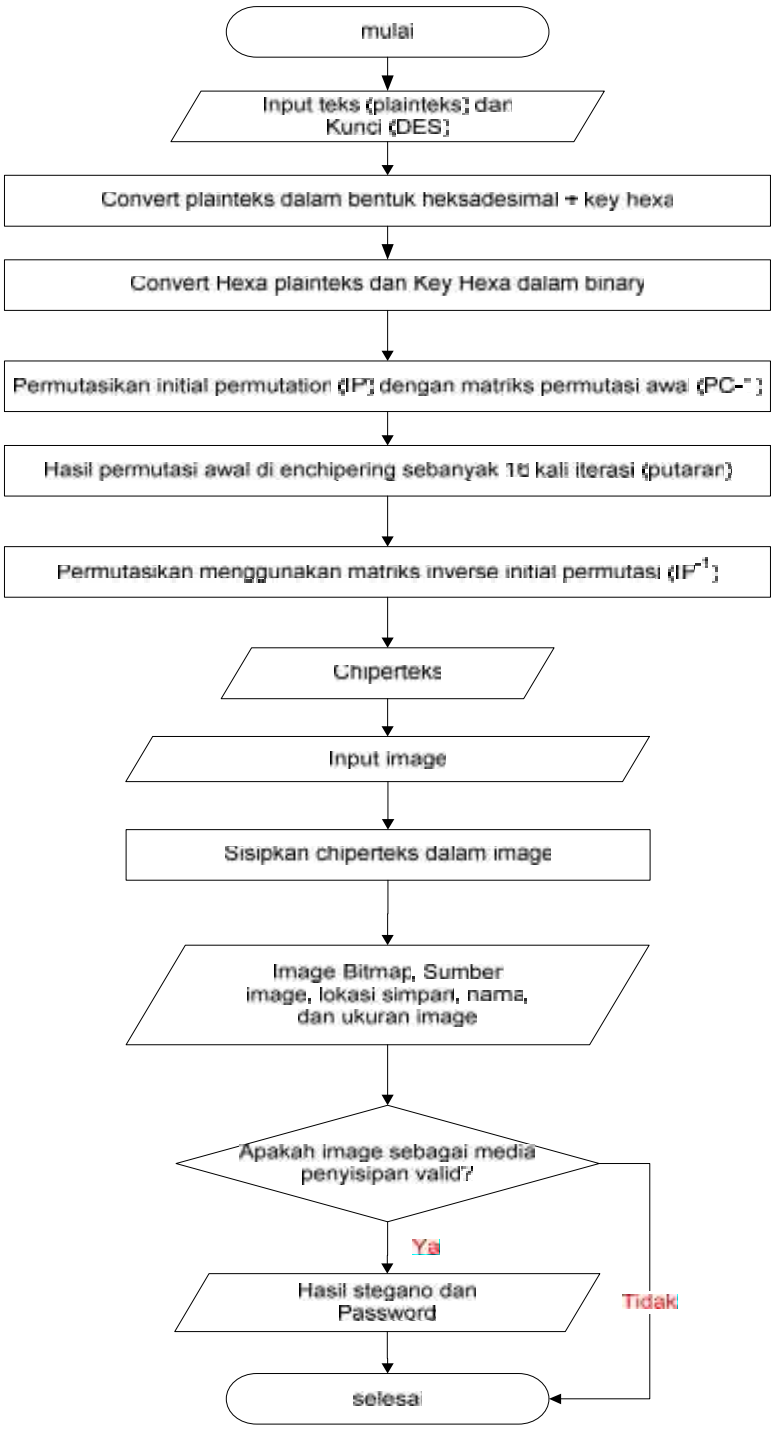
Perancangan fungsional merupakan rancangan sebuah aplikasi sistem yang meliputi *contex diagram, Data flow diagram* dan perancangan antar muka.

4.2.1.1 *Flowchart*

Proses-proses yang terjadi pada implementasi aplikasi pengamanan pesan menggunakan algoritma *Data Encryption Standard* (DES) dan *Least Significant Bit* (LSB) bisa digambarkan dengan menggunakan *flowchart* sebagai berikut :

a. *Flowchart* enkripsi dan penyisipan pesan

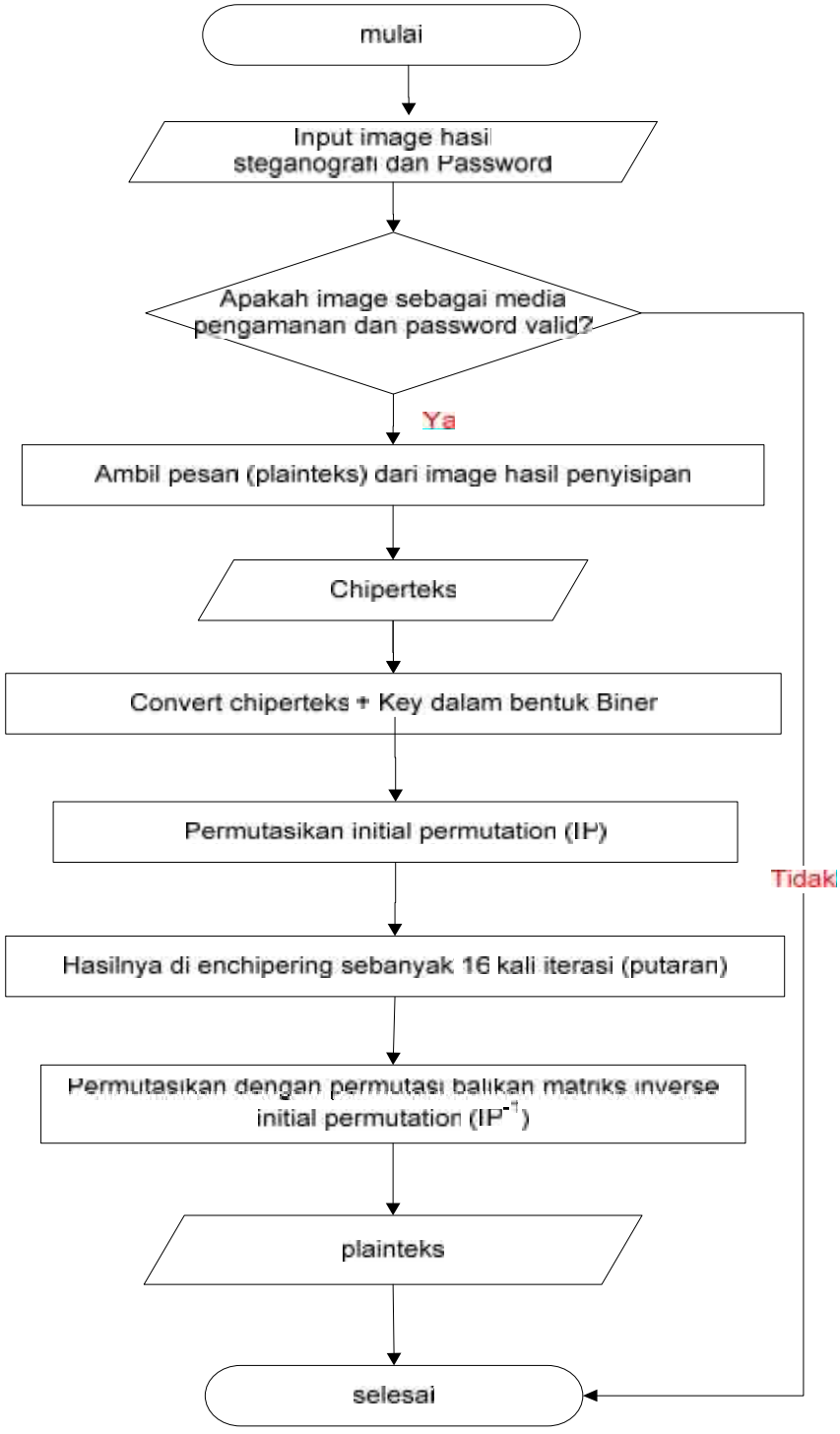
Yang berperan sebagai *user* pada proses ini adalah pengirim pesan (*sender*).



Gambar 4.5. *Flowchart* Proses Enkripsi DES dan penyisipan steganografi

b. *Flowchart* dekripsi dan pengambilan pesan

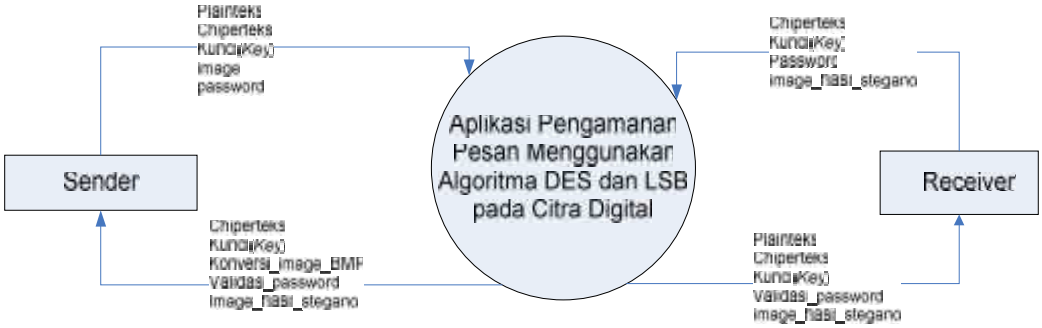
Yang berperan sebagai *user* pada proses ini adalah penerima pesan (*receiver*).



Gambar 4.6. *Flowchart* Proses Dekripsi DES dan ekstraksi steganografi

4.2.1.2 *Context diagram*

Context diagram digunakan untuk menggambarkan proses kerja aplikasi secara umum.



Gambar 4.7. *Context diagram*

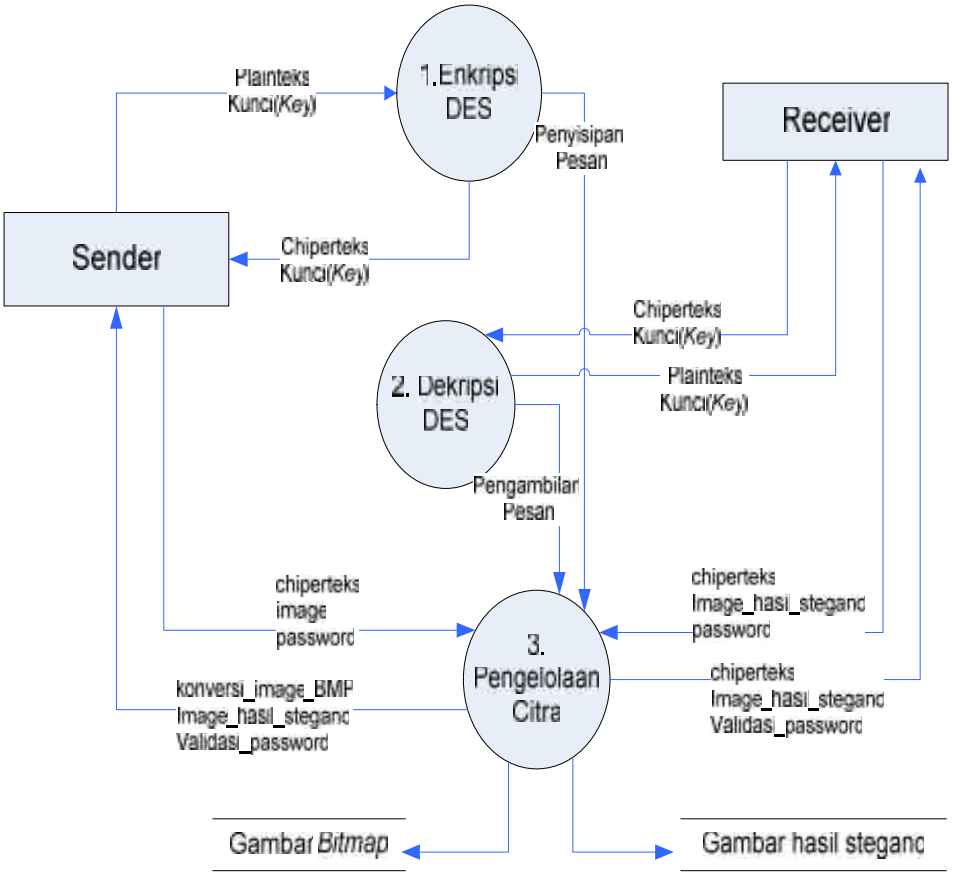
Entitas yang berinteraksi dengan aplikasi ini adalah :

1. *Sender*, memiliki peran antara lain :
   a. Memasukkan pesan yang yang ingin disampaikan
   b. Pesan dienkripsi kemudian menghasilkan sebuah chiperteks.
   c. Meng*input* gambar dalam bentuk *form*at *Bitmap*.
   d. Menyisipkan chiperteks tersebut ke dalam gambar media penampung dengan menggunakan *password*.
2. Receiver, memiliki peran antara lain :
   a. Memilih gambar *Bitmap* yang telah disisipi pesan menggunakan akses *password* yang sama dari *sender* sehingga diperoleh pesan (chiperteks).
   b. Chiperteks diproses dekripsi yang diubah menjadi plainteks untuk mendapatkan pesan aslinya.

4.2.1.3 Diagram Aliran Data (*Data flow diagram*)

Data flow diagram (DFD) digunakan untuk menggambarkan suatu aplikasi baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik data tersebut mengalir, atau lingkungan fisik data tersebut tersimpan.

a. DFD Level 1 proses pengamanan pesan menggunakan algoritma DES dan LSB pada citra digital



Gambar 4.8. DFD Level 1

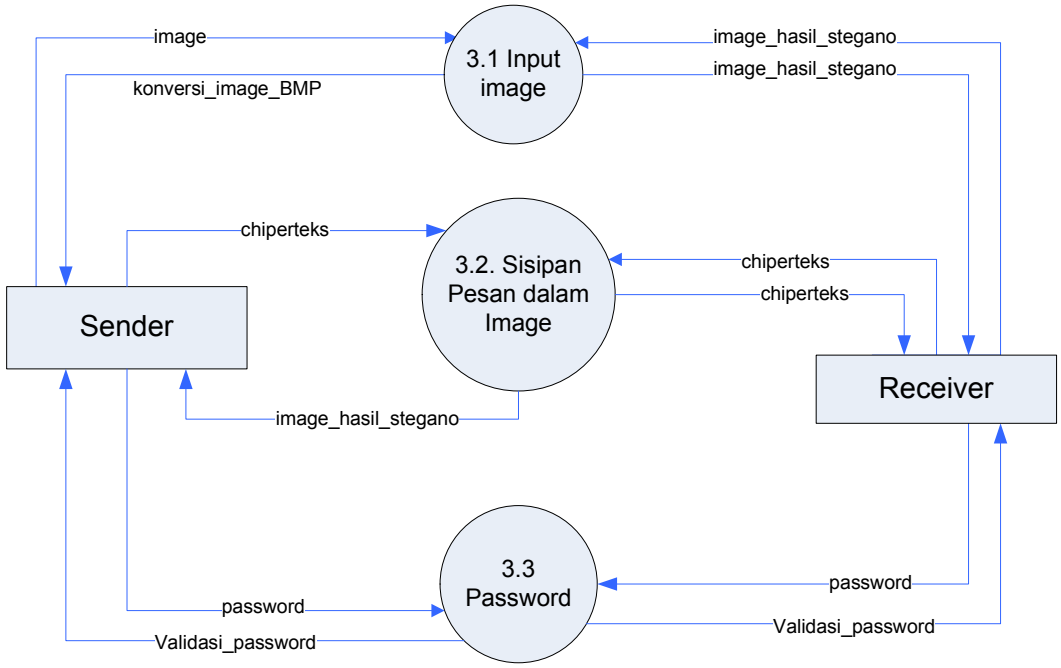
Gambar DFD Level 1 dari *Context diagram* gambar 4.8 yang dipecah menjadi 3 proses dan beberapa aliran data. Untuk keterangan masing-masing dapat dilihat pada tabel kamus data berikut ini :

Tabel 4.1 Keterangan Proses pada DFD Level 1

No	Nama proses	Masukan	Keluaran	Deskripsi
1	Enkripsi DES	- Plainteks - Kunci (*Key*)	- Chiperteks - Kunci (*Key*)	Proses plainteks menjadi chiperteks menggunakan *input*an kunci untuk proses pembangkitan kunci.
2	Dekrispi DES	- Chiperteks - Kunci (*Key*)	- Plainteks - Kunci (*Key*)	Proses chiperteks menjadi plainteks

				dengan kunci yang sama saat proses enkripsi.
3	Pengelolaan *Image*	- *Image* - konversi *image* Bmp - *Image* hasil stegano - *Password* - Validasi *Password*	- *Image* hasil stegano - *Password* - Validasi *Password*	Proses meng*input* gambar, mengkonversikan ke dalam *format Bitmap*, penyisipan chiperteks dengan *password*, dan pengambilan *password* menggunakan *password* yang sama.

b. DFD Level 2 Pengelolaan Citra



Gambar 4.9. DFD Level 2

Gambar DFD Level 2 merupakan pengembangan dari proses 3 level 1 yaitu proses pengelolaan *image*. Untuk keterangan masing-masing dapat dilihat pada tabel kamus data berikut ini :

Tabel 4.2 Keterangan Proses pada DFD Level 2

No	Nama proses	Masukan	Keluaran	Deskripsi
1	*Input image*	- *Image* - *Image* hasil stegano	- *Image* - konversi *image* Bmp - *Image* hasil stegano	Proses melakukan *input* gambar dan mengkonversikan dalam *format Bitmap*

2	Sisipan pesan dalam *image*	- chipeteks	- chipeteks - *Image* hasil stegano	Proses penyisipan dan pengambilan chiperteks dalam *image Bitmap* (*image* hasil proses stegano)
3	*Password*	- *Password*	- Validasi *Password*	melakukan *input password* yang sama pada proses penyisipan dan pengambilan pesan.

4.2.2 Perancangan *Interface* Aplikasi

Gambaran antarmuka atau *Interface* merupakan suatu sarana yang memungkinkan terjadinya interaksi antara manusia dan komputer. Oleh sebab itu, *Interface* dari sebuah perangkat lunak yang akan dibangun harus bersifat *user friendly* yang bertujuan agar pengguna (*user*) dapat mengerti dengan mudah dan memahami cara menggunakan perangkat lunak ini.

Antarmuka (*Interface*) yang dibutuhkan aplikasi yang akan dibangun ini terdiri dari beberapa bagian yang diakses *user* yang akan dirancang dan diimplementasikan dengan menggunakan bahasa pemrograman Visual Basic 6.0. perancangan ini terdapat dua proses utama yaitu *Interface* proses steganogafi (penyisipan dan pengambilan pesan dalam bitmap) dan proses kriptografi yang sesai dengan ketentuan Data Encryption Standard (proses enkripsi dan dekripsi).

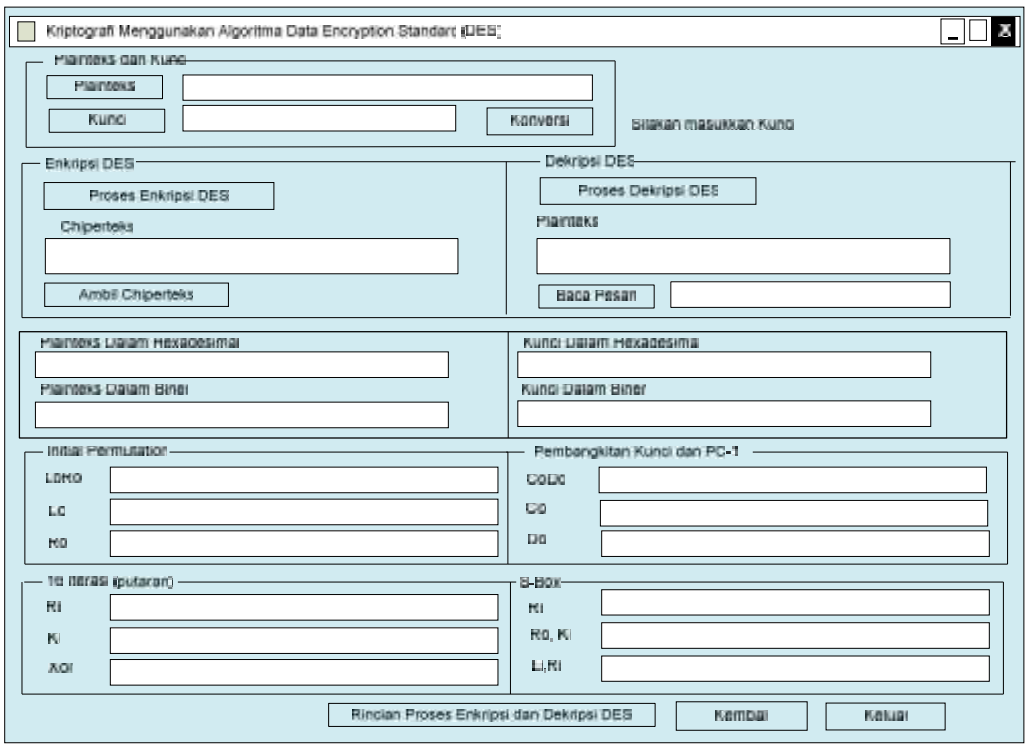
4.2.2.1 Perancangan *Interface* Menu Awal

Pada menu tampilan awal, pengguna dihadapkan pada proses rangkaian dari metode steganografi, yaitu metode Least Significant Bit (LSB) dengan meng*input*kan *image* dalam *form*at *Bitmap* dan pesan teks. Pengguna dapat melakukan penyisipan atau pengambilan gambar sesuai tab-tab menu pada aplikasi. Perancangan tampilan awal pada aplikasi ini adalah sebagai berikut :

Gambar 4.10. *Form* Tampilan Awal (Steganografi)

4.2.2.2 Perancangan *Interface* Menu Enkripsi dan Dekripsi DES

Perancangan ini merupakan rangkaian proses enkripsi dan dekripsi pesan berupa teks. *Form* ini menjelaskan tahap-tahap proses *Data Encryption Standard* secara berurutan. Menu enkripsi dan dekripsi dapat dilihat pada gambar berikut :

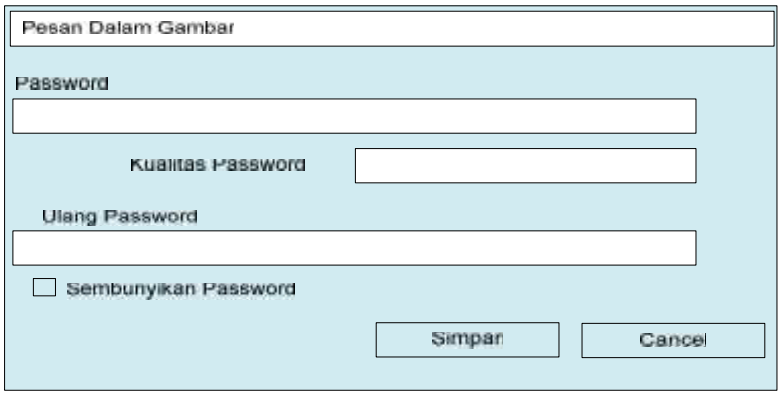


Gambar 4.11. *Form* tampilan proses enkripsi dan dekripsi DES

4.2.2.3 Perancangan *Interface* Menu *Password*

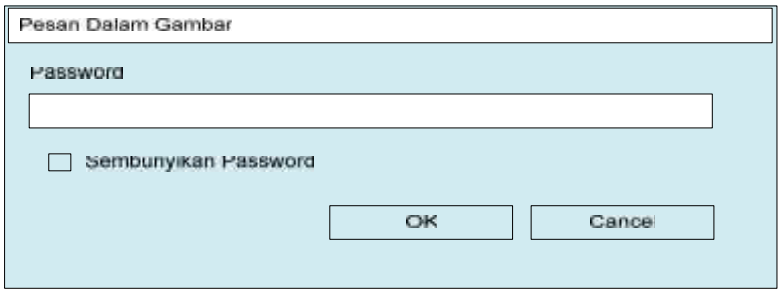
Form ini untuk meng*input*kan *password* yang merupakan bagian dari rangkaian steganografi yang berguna sebagai kata kunci akses antara *sender* dan *receiver*, yang bertujuan untuk menjaga keamanan pesan dalam gambar agar tidak dapat diambil dan dibaca oleh orang yang tidak berhak akses.

Berikut adalah *Interface* menu *password* yang dilakukan oleh *sender* pada saat melakukan penyisipan pesan setelah meng*input* gambar media penyimpanan.



Gambar 4.12. *Form password* proses penyisipan pesan dalam gambar

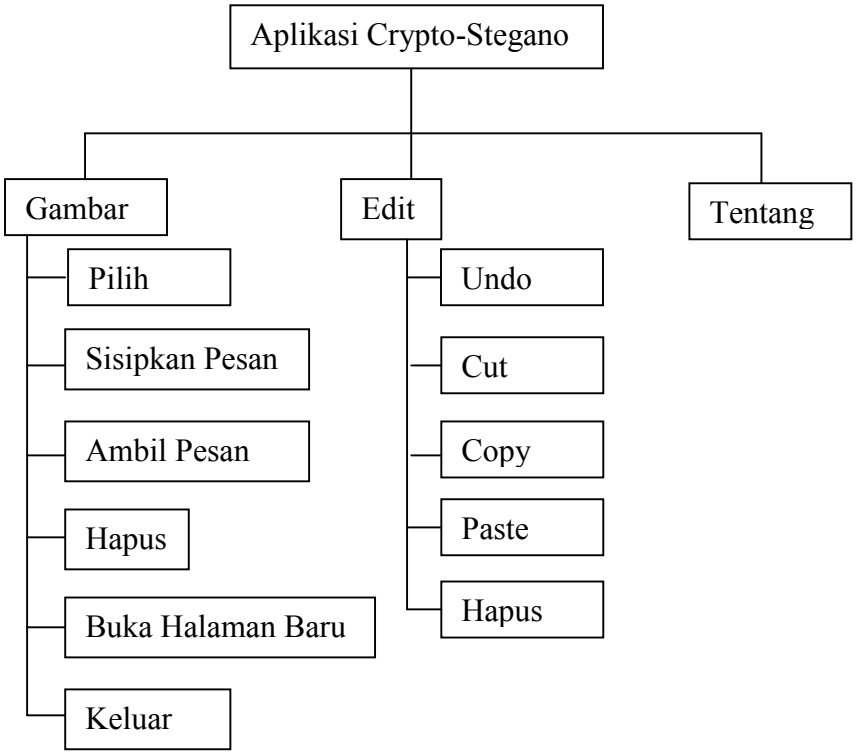
Pada proses pengambilan pesan, *receiver* juga meng*input password* yang sama sesuai dengan *sender* sebagai hak akses agar pesan dapat dibaca. Berikut adalah *Interface password* pada proses pengambilan pesan :



Gambar 4.13. *Form password* proses pengambilan pesan (chiperteks) hasil steganografi dalam gambar

4.2.3 Perancangan Struktur Menu

Rancangan aplikasi ini memiliki struktur menu dengan tiga menu utama, yaitu menu Gambar, menu Edit, dan Tentang. Struktur menunya dapat dilihat pada gambar berikut :



Gambar 4.14. Perancangan Struktur Menu Aplikasi Pengamanan Pesan *Crypto-Stego*

BAB V

IMPLEMENTASI DAN PENGUJIAN

5.1 Implementasi

Implementasi merupakan lanjutan dari tahap perancangan yaitu aplikasi yang siap dioperasikan pada keadaan yang sebenarnya, sehingga akan diketahui apakah aplikasi yang dibuat telah menghasilkan tujuan yang diharapkan.

Program aplikasi pengamanan pesan teks pada gambar *bitmap* dengan menerapkan metode algoritma *Data Encryption Standard* (DES) dan *Least Significant Bit (LSB)* yang memanfaatkan perangkat lunak *Microsoft Visual Basic 6.0*.

5.1.1 Alasan Pemilihan Perangkat Lunak

Perangkat lunak yang digunakan dalam implementasi steganografi ini adalah *Microsoft Visual Basic 6.0* untuk penanganan antar mukanya berdasarkan beberapa pertimbangan yaitu:

Microsoft Visual Basic 6.0 hampir dapat memanfaatkan seluruh kemudahan dan kecanggihan yang dimiliki oleh sistem operasi *Windows*. Apalagi objek-objek yang disediakan mudah digunakan sehingga dapat dibuat aplikasi yang sesuai dengan tampilan dan cara kerja *Windows*.

5.1.2 Batasan Implementasi

Batasan implementasi pada penulisan tugas akhir ini adalah :

1. Aplikasi ini tidak menyimpan *password* proses steganografi, sehingga apabila *user* lupa atau kehilangan *password* maka tidak dapat mengambil kembali pesan yang telah disisipkan.
2. Implementasi ini tidak membahas proses pengiriman dan penerimaan pesan hasil kripto-stegano.

5.1.3 Lingkungan Implementasi

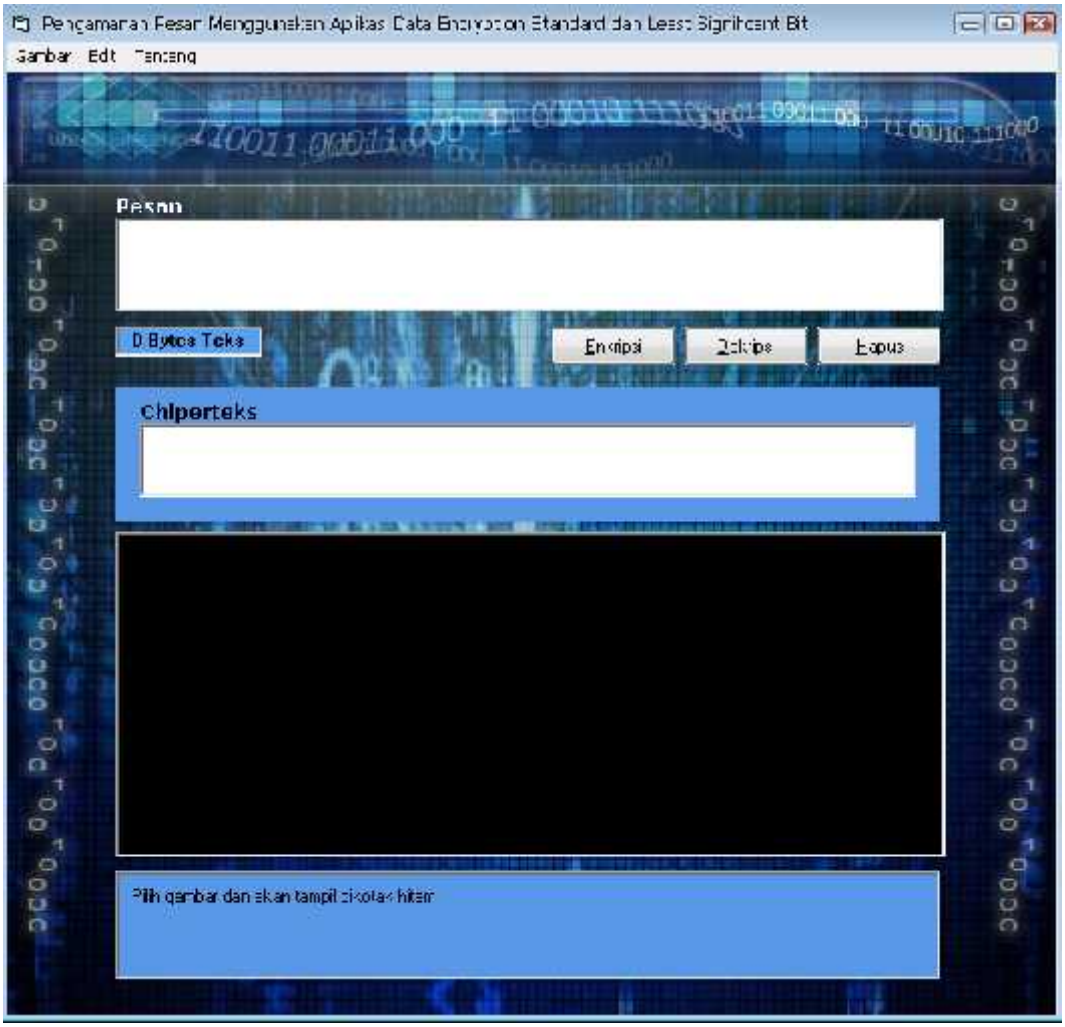
Pada aplikasi *crypto-stegano* ini yang menjadi kebutuhan penting berupa kebutuhan inputan data dan hasil output yang memiliki arti penting dalam teknik pengamanan pesan juga mempunyai kebutuhan lingkungan implementasi yang terdiri kebutuhan aplikasi dari perangkat keras dan perangkat lunak. Berikut adalah spesifikasi dari masing-masing perangkat :

1. Lingkungan Perangkat Keras
    a. *Processor* : Intel Core Duo 2.0 GHz
    b. *Memory* : 512 MB
    c. *Harddisk* : 120 GB
    d. *Monitor*
    e. *Keyboard*
    f. *Mouse*
2. Lingkungan Perangkat Lunak
    a. Sistem Operasi : *Windows XP Professional*
    b. *Tools Image Viewer* : *Windows Photo Viewer*
    c. Bahasa pemrograman : *Ms. Visual Basic 6.0*

5.1.4 Tampilan Aplikasi

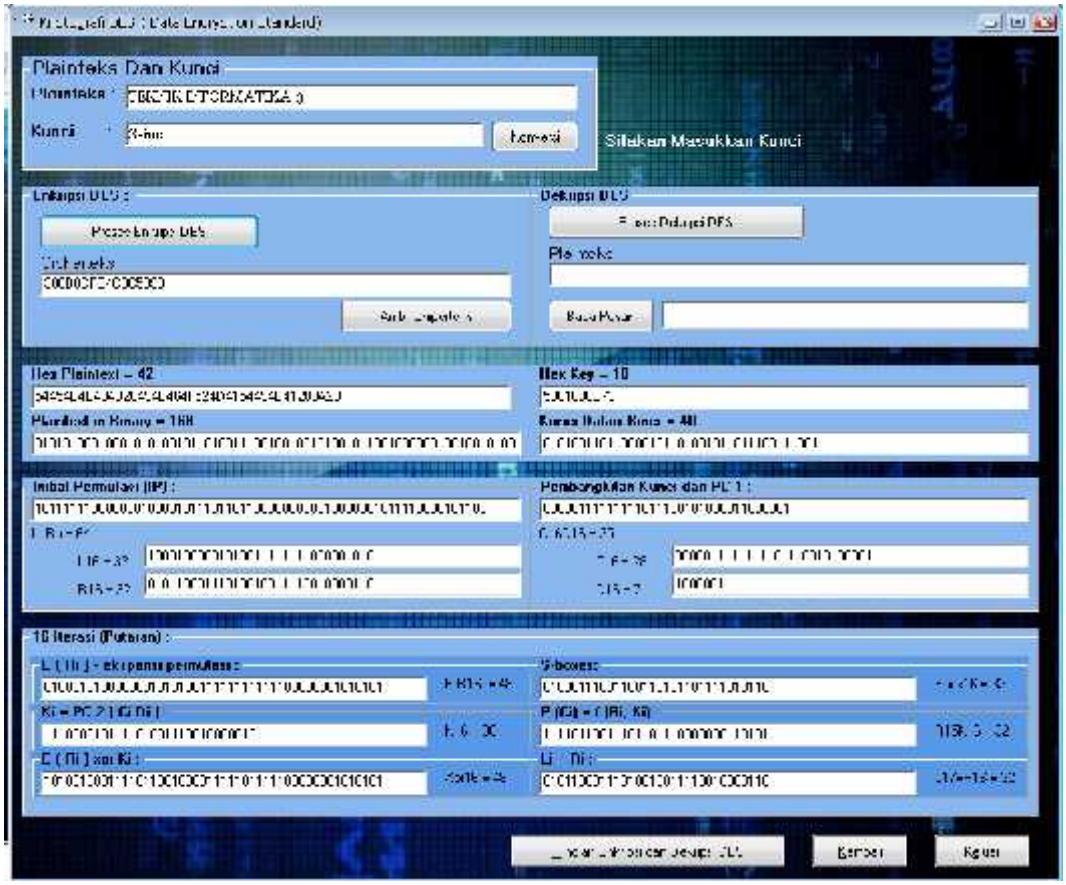
Aplikasi pada tugas akhir ini dirancang untuk membantu mengamankan pesan teks dalam gambar dengan format *bitmap* 24 bit. Tampilan antar muka (interface) dari aplikasi keamanan data menggunakan teknik gabungan kriptografi dan steganografi ini secara umum diperlihatkan melalui tampilan menu utama pada gambar 5.1.

Aplikasi ini terdiri dari beberapa *Form* yang mempunyai aturan tertentu yang harus dijalankan secara berurutan.



Gambar 5.1 Tampilan Menu Utama Aplikasi Pengamanan Pesan Menggunakan Algoritma DES dan LSB Pada Citra Digital

Menu berikutnya merupakan menu kriptografi Data Encryption Standard untuk melakukan proses enkripsi dan deskripsi. Pada proses enkripsi akan menghasilkan chiperteks kemudian disisipkan dalam gambar *Bitmap.*



Gambar 5.2 Tampilan Enkripsi – Dekripsi DES

5.2 Pengujian Aplikasi

Pengujian dilakukan dengan memperlihatkan penyisipan data teks kedalam gambar *bitmap*. Kemudian pembuktian dilakukan untuk membuktikan bahwa teks yang dihasilkan setelah enkripsi adalah sesuai dengan masukan (*input*). Pengujian aplikasi yang sesuai dengan lingkungan implementasi meliputi :

1. Pengujian menggunakan *Black Box*.
2. Pengujian berdasarkan *fidelity.*
3. Pengujian menggunakan *Tools StegSpy* 2.1

5.2.1 Pengujian Aplikasi Menggunakan *Black Box*

Pengujian dilakukan dengan melihat *interface* masing-masing form yang ada dalam setiap urutan proses gabungan kriptografi dan steganografi pada gambar *Bitmap.* Pengujian *black box* merupakan pengujian modul dengan hasil

pengujian implementasi aplikasi secara detail mengenai *item-item* yang terdapat pada setiap tampilan proses menyisipkan dan mengambil pesan dalam gambar.

5.2.1.1 Pengujian Modul Enkripsi

Prekondisi

1. Sudah ada teks yang diinput sebagai pesan informasi yang akan disisipkan dalam gambar.

Tabel 5.1 Tabel Butir Uji Pengujian Enkripsi Pesan

Deskripsi	Prekondisi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
Pengujian tahap 1 Proses enkripsi teks	Tampilan layar menu utama, Ada textbox untuk pesan yang akan disisipkan, fasilitas option, enkripsi, dekripsi dan hapus. Kemudian masukkan kunci.	1.Tekan tombol "Enkripsi" untuk mengenkripsikan pesan teks 2.Tekan tombol "Konversi" untuk mengkonversikan pesan (plainteks) dan kunci dari ASCII ke dalam bentuk heksadesimal. 3.Tekan tombol "Proses Enkripsi DES" untuk mengkonversikan plainteks dan kunci	Pesan teks (plainteks) dan kunci.	Chiperteks hasil enkripsi dan tidak ada instruksi *error*	Chiperteks dalam bentuk heksadesimal dan tidak ada instruksi *error*	Chiperteks dalam bentuk heksadesimal dan tidak ada instruksi *error*	Diterima

Tabel 5.1 Tabel Butir Uji Pengujian Enkripsi Pesan (Lanjutan)

Deskripsi	Prekondisi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
		yang sudah dalam bentuk heksadesimal kedalam bentuk biner dan langkah-langkah proses DES secara beurutan sehingga menghasilkan chiperteks.					

5.2.1.2 Pengujian Modul Penyisipan Pesan Dalam Gambar

Prekondisi

1. Sudah ada pesan berupa chiperteks hasil enkripsi dalam bentuk heksadesimal yang akan disisipkan.
2. Tahap 1 telah dilalui tidak ada instruksi *error*

Tabel 5.2 Tabel Butir Uji Pengujian Penyisipan Pesan Dalam Gambar

Deskripsi	Prekondisi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
Pengujian tahap 3 Proses Penyisipan pesan teks dalam gambar	Tampilan layar menu utama, ada fasilitas menu penyisipan dan		Chiperteks hasil enkripsi yang akan disisipkan ke dalam gambar, Gambar sebagai media penyimpa	Chiperteks yang akan disisipkan dalam gambar, gambar media penyimpanan yang berhasil dikonversi ke bitmap, dan tidak ada instruksi	Chiperteks telah berhasil disisipkan dalam gambar yang telah dipilih sebagai media penyimpa nannya, gambar	Chiperteks telah berhasil disisipkan dalam gambar yang telah dipilih sebagai media penyimpa nannya, gambar	Diterima

Tabel 5.2 Tabel Butir Uji Pengujian Penyisipan Pesan Dalam Gambar (Lanjutan)

Deskripsi	Prekondisi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
	pengamabilan pesan dalam gambar. Tahap 1telah dilalui tidak ada instruksi *error*	"Sisipkan Pesan" untuk Menyisipkan pesan berupa chiperteks 3.setelah berhasil disisipkan akan tampil pesan dan *form password*	nan, gambar dikonversi kedalam format *Bitmap*, *Password* untuk keamanan.	*error*	berhasil dikonversi ke bitmap dan ukuran gambar berubah, dan tidak ada instruksi *error*	berhasil dikonversi	

5.2.1.3 Pengujian Modul Memasukkan *Password* Setelah Proses Penyisipan

Prekondisi

1. Sudah ada pesan dalam gambar yang telah berhasil disisipkan dan gambar telah dikonversi kedalam *bitmap*.
2. Tahap 1 dan 3 telah dilalui tidak ada instruksi *error*

Tabel 5.3 Tabel Butir Uji Pengujian Memasukkan *Password* Setelah Proses Penyisipan

Deskripsi	Prekondisi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
Pengujian tahap 4 Proses Memasukkan *password*	Tahap 1 dan 3 telah dilalui tidak ada instruksi	1.Setelah tampil *form password* untuk memasukkan kata kunci 2.Silahkan *input*	*Password*	Jika *password* benar, penyisipan berhasil dilakukan. jika *password* salah atau tidak cocok, tampil pesan	Jika *password* benar, tampil pesan proses penyisipan pesan dalam gambar	Jika *password* benar, tampil pesan proses penyisipan pesan dalam gambar	Diterima

Tabel 5.3 Tabel Butir Uji Pengujian Memasukkan *Password* Setelah Proses Penyisipan (Lanjutan)

Deskripsi	Prekondisi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
	error	*password*nya. 3. Klik tombol Simpan		kesalahan dan tidak ada instruksi *error*.	berhasil. jika *password* salah atau tidak cocok, tampil pesan kesalahan dan tidak ada instruksi *error*.i *error*.	berhasil. jika *password* salah atau tidak cocok, tampil pesan kesalahan dan tidak ada instruksi *error*.	

5.2.1.4 Pengujian Modul Mengambil Pesan Teks dalam Gambar *Bitmap*

Prekondisi

1. Sudah ada chiperteks proses penyisipan dalam gambar *Bitmap* yang merupakan hasil stegano.

Tabel 5.4 Tabel Butir Uji Pengujian Mengambil Pesan Teks dalam Gambar *Bitmap*

Deskripsi	Prekondisi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
Pengujian tahap 5 Proses pengambilan pesan teks dalam gambar *Bitmap* hasil stegano.	Sudah dipilih gambar Bitmap hasil stegano yang menjadi media	1.Klik tombol "Gambar" kemudian klik tombol "Pilih" untuk menginput gambar *Bitmap* hasil	Gambar Bitmap hasil stegano yang sudah dipilih, dam masukkan *password* yang sama pada	Pesan dalam gambar hasil stegano berhasil diambil, *password* yang dimasukkan cocok, dan tidak ada instruksi *error*.	Pesan dalam gambar hasil stegano berhasil diambil, dan tampil pesan pengambilan pesan telah berhasil.	Pesan dalam gambar hasil stegano berhasil diambil, dan tampil pesan pengambilan pesan	Diterima

Tabel 5.4 Tabel Butir Uji Pengujian Mengambil Pesan Teks dalam Gambar *Bitmap* (Lanjutan)

Deskripsi	Prekondisi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
	penyimpanan pesan teks berupa chiperteks dalam heksadesimal	stegano. 2. Tampil *form Password* untuk pengmabilan pesan dalam gambar.	proses penyisipan	Pesan dalam gambar hasil stegano berhasil diambil, *password* yang dimasukkan cocok, dan tidak ada instruksi *error*.	*Password* yang dimasukkan cocok dengan proses penyisipan dan akn dilakukan proses dekripsi, dan tidak ada instruksi *error*.	telah berhasil. *Password* yang dimasukkan cocok dengan proses penyisipan dan akn dilakukan proses dekripsi, dan tidak ada instruksi *error*.	

5.2.1.5 Pengujian Modul Memasukkan *Password* Untuk Pengambilan Pesan Teks dalam Gambar *Bitmap* Hasil Stegano.

Prekondisi

1. Sudah ada pilihan gambar *bitmap* hail stegano.

Tabel 5.5 Tabel Butir Uji Pengujian Memasukkan *Password* Untuk Pengambilan Pesan Teks dalam Gambar *Bitmap* Hasil Stegano.

Deskripsi	Prekondisi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
Pengujian tahap 5 Proses memasukkan *password* untuk pengambilan pesan teks dalam	Tahap 5 berhasil dilakukan dan tidak	1.Setelah pilih gambar bitmap hasil stegano, tampil *form password*.	*Password* yang sama pada saat proses penyisipan.	*Password* sama dan konfirmasinya berhasil dengan proses penyisipan dan tidak ada instruksi *error*	*Password* berhasil dan pesan teks dapat diambil dan tidak ada instruksi *error*	*Password* berhasil dan pesan teks dapat diambil dan tidak ada instruksi *error*	Diterima

Tabel 5.5 Tabel Butir Uji Pengujian Memasukkan *Password* Untuk Pengambilan Pesan Teks dalam Gambar *Bitmap* Hasil Stegano (Lanjutan)

Deskripsi	Prekondisi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
gambar *bitmap* hasil stegano..	ada instruksi *error*.	2.Silahkan input *password* yang sama pada proses penyisipan 3. Klik "OK" untuk konfirmasi pesan sehingga dapat melakukan proses dekripsi.	*Password* yang sama pada saat proses penyisipan.	*Password* sama dan konfirmasinya berhasil dengan proses penyisipan dan tidak ada instruksi *error*	*Password* berhasil dan pesan teks dapat diambil dan tidak ada instruksi *error*	*Password* berhasil dan pesan teks dapat diambil dan tidak ada instruksi *error*	Diterima

5.2.1.6 Pengujian Modul Dekripsi

Prekondisi

1. Sudah ada chiperteks hasil enkripsi yang telah diambil dari gambar.

Tabel 5.6 Tabel Butir Uji Pengujian Dekripsi Pesan

Deskripsi	Prekondisi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
Pengujian tahap 2 Proses dekripsi teks	Tampilan layar menu utama, Ada textbox sudah berisi.	1.Klik tombol "Dekripsi" untuk mendekripsikan chiperteks 2.Klik tombol "Proses Dekripsi DES" untuk	Chiperteks hasil enkripsi dalam bentuk heksadesimal.	Plainteks (Pesan asli), kunci dan tidak ada instruksi *error*	Pesan asli dan tidak ada instruksi *error*	Pesan asli dan tidak ada instruksi *error*	Diterima

Tabel 5.6 Tabel Butir Uji Pengujian Dekripsi Pesan (Lanjutan)

Deskripsi	Prekondisi	Prosedur Pengujian	Masukan	Keluaran yang Diharapkan	Kriteria Evaluasi Hasil	Hasil yang didapat	Kesimpulan
	chiperteks, fasilitas option, enkripsi, dekripsi dan hapus	mengkonversikan chiperteks yang sudah dalam bentuk heksadesimal kedalam bentuk biner dan langkah-langkah proses dekripsi secara beurutan untuk membalikkan chiperteks ke pesan asli.					

5.2.2 Pengujian Aplikasi Berdasarkan *Fidelity*

Pada pengujian ini dilakukan yang bertujuan untuk mengetahui mutu citra objek penyimpanan apakah mengalami perubahan kapasitas atau tidak. Pengujian *fidelity* dilakukan dengan melihat perubahan besar file gambar pada saat sebelum dan sesudah disisipkan pesan teks. Pengujian besar file dilakukan pada 5 gambar *bitmap* yang berbeda dengan 3 kali percobaan pada masing-masing *bitmap.* Hasil pengujian dapat dilihat pada tabel dibawah ini dan lampiran E.

Tabel 5.7 Uji Pengujian Aplikasi Berdasarkan *Fidelity*

Gambar Bitmap	Karakter Teks	Ukuran Citra (*pixel* x *pixel*)	Besar Kapasitas Bitmap	
			Awal (sebelum)	Akhir (setelah)
0009772 10001. bmp	TEKNIK INFORMATIKA	3024 x 2005	17,3 MB	17,3 MB
IMG_27 59. bmp	TEKNIK INFORMATIKA	3648x 2736	28,5 MB	28,5 MB
_DSC00 95. bmp	TEKNIK INFORMATIKA	4288 x 2848	34,9 MB	34,9 MB

Dapat disimpulkan bahwa besar kapasitas gambar *bitmap* hasil steganografi (lampiran E) tidak mengalami perubahan, karena proses penyisipan biner teks ke dalam biner *bitmap* menggunakan metode penggantian bit terakhir tidak mempengaruhi kapasitas gambar dan tidak diketahui secara kasat mata dengan jelas. Hal ini merupakan salah satu kelebihan dari metode Least Significant Bit.

5.2.3 Pengujian Aplikasi Menggunakan *Tools StegSpy* 2.1

Pengujian dilakukan sebanyak lima kali pengujian dengan menggunakan aplikasi *StegSpy V 2.1* produksi *Spy Hunter* pada *website* www.spy-hunter.com yang merupakan steganalisis *tools* untuk mendeteksi keberadaan pesan rahasia dalam *bitmap* hasil steganografi. Hasil pengujian dapat dilihat pada tabel dibawah ini:

Tabel 4.9 Tabel Pengujian Keberadaan Teks Dalam Citra *Bitmap*

No	File	Hasil
1	000977210001. Bmp	Terdeteksi
2	_DSC0095. bmp	Terdeteksi
3	DSC01473. bmp	Terdeteksi
4	_DSC0105. bmp	Terdeteksi
5	IMG_2759. Bmp	Terdeteksi

Dari pengujian menggunakan *StegSpy V 2.1* pada *bitmap* hasil steganografi, maka data rahasia yang disisipkan pada kelima file tersebut

menggunakan metode *LSB* dapat terdeteksi. Pengujian ini dapat dilihat pada lampiran F.

5.3 Kesimpulan Pengujian

Setelah membandingkan antara hasil perancangan dan hasil yang didapat, maka dapat dilihat bahwa pengamanan pesan berupa teks dengan gabungan teknik kriptografi dan steganografi menggunakan metode Data Encryption Standard (DES) dan Least Significant Bit (LSB) pada citra digital *bitmap* dapat dilakukan dengan sempurna. Tampilan aplikasi yang dihasilkan bersifat *user friendly* ketika diuji coba kepada beberapa *user*. Adapun yang dapat disimpulkan dari beberapa pengujian sebagai berikut:

1. Besar *bitmap* hasil steganografi tidak mengalami perubahan, karena proses penyisipan biner teks ke dalam biner *bitmap* menggunakan metode penggantian bit terakhir sehingga kapasitas *bitmap* sebelum dan sesudah disteganografi tidak mengalami perubahan yang berarti.
2. Pengujian pesan yang akan dienkripsi dari teks dan kunciyang digunakan sebagai pembangkitan kunci dalam proses DES dapat dikonversi kedalam heksadesimal dan biner dan sesuai dengan tabel ASCII.
3. Pengujian pesan teks menggunakan metode DES berhasil dilakukan sesuai tepat dengan alur atau langkah-langkahnya sehingga menghasilkan chiperteks yang berupa heksadesimal.
4. Pengujian dilakukan dengan melihat data yang disembunyikan harus tahan terhadap manipulasi yang dilakukan pada citra penampung. Dari hasil pengujian diketahui bahwa proses pemotongan dapat merusak karakter teks yang berada dalam *bitmap,* karena terjadinya perubahan letak biner. Hal ini terbukti dari proses ekstraksi hasil yang gagal dilakukan karena *bitmap* telah rusak.
5. Pengujian dilakukan dengan menjalankan aplikasi ekstraksi hasil, dapat diambil kesimpulan bahwa pengungkapan data kembali berhasil dan pesan teks dapat diambil.

6. Pesan yang akan diambil dari bitmap dapat dilanjutkan ke proses dekripsi yang bertujuan untuk mengembalikan pesan chiperteks ke bentuk semula (plainteks asli) melalui proses dekripsi DES yang sesuai.
7. *Password* yang diinput pada saat proses penyisipan dan pengambilan pesan teks adalah sama yang menunjukkan bahwa satu hak kepemilikan.
8. Pengujian dilakukan sebanyak lima kali pengujian dengan menggunakan aplikasi *StegSpy V 2.1,* data rahasia yang disisipkan pada kelima gamabar *bitmap* tersebut menggunakan metode *LSB* dapat terdeteksi.

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan analisa, perancangan dan implementasi pada aplikasi keamanan pesan teks menggunakan teknik gabungan kriptografi dan steganografi pada citra digital *bitmap* dengan metode *Data Encryption Standard (DES)* dan *Least Significant Bit (LSB)*, diperoleh kesimpulan dari tugas akhir yang antara lain adalah :

1. Proses pengamanan pesan pada citra digital aman dan tidak diketahui secara kasat mata, karena besar *bitmap* hasil steganografi tidak mengalami perubahan setelah proses penyisipan biner teks ke dalam biner *bitmap* menggunakan metode penggantian bit terakhir sehingga kapasitas *bitmap* sebelum dan sesudah disteganografi tidak mengalami perubahan yang berarti yang telah diuji berdasarkan *fidelity*.
2. Pengujian dilakukan dengan melihat data yang disembunyikan harus tahan terhadap manipulasi yang dilakukan pada citra penampung. Dari hasil pengujian diketahui bahwa proses pemotongan dapat merusak karakter teks yang berada dalam *bitmap,* karena terjadinya perubahan letak biner. Hal ini terbukti dari proses ekstraksi hasil yang gagal dilakukan karena *bitmap* telah rusak.
3. Pengujian dilakukan dengan menjalankan aplikasi ekstraksi hasil, dapat diambil kesimpulan bahwa pengungkapan data kembali berhasil dan pesan teks dapat diambil.
4. Pengujian dilakukan sebanyak lima kali pengujian dengan menggunakan aplikasi *StegSpy V 2.1,* data rahasia yang disisipkan pada kelima gamabar *bitmap* tersebut menggunakan metode *LSB* dapat terdeteksi.

6.2 Saran

Agar penulisan laporan penelitian tugas akhir ini bermanfaat dan berdaya guna dimasa sekarang dan yang akan datang, maka penulis memberikan beberapa hal yang disaran dengan menerapkan metode ini sebagai berikut:

1. Aplikasi ini hanya terbatas pada citra dengan format *bitmap*, untuk itu disarankan agar dapat dikembangkan dengan format lain agar ruang lingkup penggunaan metode steganogarfi ini lebih luas, seperti audio, video, atau document file.
2. Pada tugas akhir ini pengujian serangan (*attack*) hanya dilakukan dengan mencoba kemungkinan penggunaan kunci untuk itu agar dapat dikembangkan motode serangan (*attack*) yang lainnnya guna menguji kekuatan algoritma *Data Encryption Standard* dan *Least Significant Bit*.
3. Diharapkan apabila ada penelitian lanjutan mengenai penyandian citra menggunakan metode aplikasi ini yang dirancang agar dapat mengatasi masalah peng*edit*an teks*,* sehingga apabila ada sebuah teks yang telah dienkripsi lalu dirotasi aplikasi tetap dapat membalikkan teks menjadi teks aslinya.
4. Pada pengembangan aplikasi ini dapat dilakukan dengan metode kriptografi *Triple DES (3-DES)*, yang merupakan pengembangan teknik DES yaitu 3 kali putaran DES.

DAFTAR PUSTAKA

Arifin, Rachmad. *Data Encryption Standards (DES)*. Teknik Informatika. Sekolah Teknologi Elektro dan Informatika. Institut Teknologi Bandung. 2009.

Ariyus, Dony. Pengantar Ilmu Kriptografi : Teori, Analisis, dan Implementasi, Andi Yogyakarta, Yogyakarta. 2008

_____. Kriptografi:Keamanan Data dan Komunikasi, Graha Ilmu, Yogyakarta. 2006

Astrianto, Stefanus. Analisa Algoritma *Block Chiper* dalam Penyandian *DES* dan pengembangannya. 2007.

Dian Dari Hapsari, Lintang Yuniar Banowosari. Aplikasi Video Steganografi dengan Metode LSB. Konferensi Nasional Sistem dan Informatika 2009, Bali, November 14, 2009.

Felix, Fidens. Dasar Kriptografi, http://www.ilmukomputer.com, 2006

Henry. "Video Steganography" http://www.cert.or.id/~budi/courses/security/2006/henry_report.pdf, 2007

Munir, Rinaldi. Diktat Kuliah IF5054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung. 2004

_____. Kriptografi, Informatika, Bandung.2006

Neil F. Johnson, Sushil Jajodia, "*Steganography: Seeing the Unseen"*.1995

Roman, Yusrian. Audio Steganografi. 2007

Setiawan, Rachmansyah Budi. Penggunaan Kriptografi dan Steganografi Berdasarkan Kebutuhan dan Karakteristik Keduanya. , Program Studi Teknik Informatika, Institut Teknologi Bandung. 2009.

Sutanto, Arnold Nugroho. Studi dan Analisis Teknik-Teknik Pendeteksian Steganografi Dengan Metode LSB dalam Media Gambar, Program Studi Teknik Informatika, Institut Teknologi Bandung. 2007

Wikipedia."Steganography" , http://en.wikipedia.org/wiki/Steganography, 2008